

# **SANDIA REPORT**

SAND2014-18511

Unlimited Release

Printed October 2014

## **A Thermo-Optic Propagation Modeling Capability**

Karl N. Schrader

Ronald L. Akau

Prepared by Karl N. Schrader  
Sandia National Laboratories  
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



**Sandia National Laboratories**

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831

Telephone: (865) 576-8401  
Facsimile: (865) 576-5728  
E-Mail: [reports@adonis.osti.gov](mailto:reports@adonis.osti.gov)  
Online ordering: <http://www.osti.gov/bridge>

Available to the public from

U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Rd.  
Springfield, VA 22161

Telephone: (800) 553-6847  
Facsimile: (703) 605-6900  
E-Mail: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



# **A Thermo-Optic Propagation Modeling Capability LDRD 165571 Final Report**

Karl N. Schrader  
Optics and Sensor Engineering Department

Ronald L. Akau  
Thermal Sciences and Engineering Department

Sandia National Laboratories  
P.O. Box 5800  
Albuquerque, New Mexico 87185-MS0406

## **Abstract**

A new theoretical basis is derived for tracing optical rays within a finite-element (FE) volume. The ray-trajectory equations are cast into the local element coordinate frame and the full finite-element interpolation is used to determine instantaneous index gradient for the ray-path integral equation. The FE methodology (FEM) is also used to interpolate local surface deformations and the surface normal vector for computing the refraction angle when launching rays into the volume, and again when rays exit the medium.

The method is implemented in the Matlab<sup>TM</sup> environment and compared to closed-form gradient index models. A software architecture is also developed for implementing the algorithms in the Zemax<sup>TM</sup> commercial ray-trace application. A controlled thermal environment was constructed in the laboratory, and measured data was collected to validate the structural, thermal, and optical modeling methods.

## **ACKNOWLEDGMENTS**

**This work was funded under LDRD Project Number 165571 and Title "A Thermo-Optic Propagation Capability for Reducing Design-Cycle Time, Improving Performance Margins, and Lowering Realization Costs".**



# CONTENTS

1.	Introduction .....	9
2.	The Thermo-Optic analysis Process.....	11
2.1	Optical System Overview .....	11
2.2	Classical Thermo-Optic Analysis .....	11
2.3	Finite-Element Ray Trace Analysis.....	12
3.	Ray tracing in the finite element domain .....	13
3.1	Ray-Trace Equations in the Finite Element Domain.....	13
3.2	Ray-Trace Algorithm.....	13
3.3	FE optical surface interpolation error .....	14
3.4	Ray intercept calculation .....	16
3.4.1	Intercept of the nominal surface .....	17
3.4.2	Intercept of the distorted/displaced surface .....	18
4.	Implementation in the Zemax User-Defined Surface .....	19
4.1	The User-Defined Surface (UDS) API .....	19
4.2	Zemax ray-trace requests.....	20
4.3	Using the FE surface in a Zemax lens file.....	21
4.3.1	Modeling an FE singlet in a lens file .....	21
4.3.2	Typical use-case scenario .....	23
4.3.3	Custom Analyses .....	25
4.3.4	UDS Software Status .....	26
5.	Model Validation And Testing.....	27
5.1	Test Description.....	27
5.2	Interferometer Test Setup .....	27
5.3	Temperature and IR Measurements.....	30
5.3.1	NBK7 Optical Material.....	30
5.3.2	Fused Silica (FS) Optical Material .....	32
5.4	Comparison of Thermal Measurements and Model Predictions .....	34
5.5	Interferometry Measurements.....	38
5.6	SIERRA/Aria Thermal Displacement Predictions for TOP LDRD .....	39
6.	Conclusions .....	44
6.1	Significant achievements .....	44
6.2	Follow-on effort.....	44
6.2.1	Software Development .....	44
6.2.2	Software Validation .....	44
6.3	Future Impact: Coupled Thermo-Optic Transient Analysis .....	45
	References.....	46
	Appendix A: Theoretical Basis.....	48
	Appendix B: ANSYS Analysis and Model Preparation for Output to ZEMAX.....	60
	Appendix C: Zemax Compatible Sierra Mechanics FEM Model .....	66

Appendix D: Gravity & Mount Induced Deformation of Optic during Thermal Testing .....	68
Appendix E: Sample FEM File Format.....	72
Distribution .....	75

## FIGURES

Figure 1. The Zemax Surface Properties dialog. ....	21
Figure 2. Open-File dialog for the FE Data File.....	22
Figure 3. Dialog showing requirement for a second UDS surface. ....	22
Figure 4. Lens file listing for the first surface of the FE singlet. ....	22
Figure 5. Complete lens file listing for the FE singlet, showing front and back surface. ....	23
Figure 6. The FE Data tab of the <i>Surface Properties</i> dialog. ....	24
Figure 7. Custom analyses inserted into the Zemax main menu. ....	25
Figure 8. Rendering of the finite-element singlet in its local coordinate frame. ....	26
Figure 9. Four-surface interferometer test setup.....	28
Figure 10. Infrared camera test setup, (a) Two-axis mount removed, (b) black aluminum cover. .....	28
Figure 11. Thermocouple locations on N-BK7 window. ....	29
Figure 12. Thermocouple locations on FS window.....	29
Figure 13. IR images of NBK7 back side.....	31
Figure 14. IR images of NBK7 front side.....	32
Figure 15. IR images of FS back side.....	33
Figure 16. IR images of FS front side.....	34
Figure 17. ANSYS window geometry model and 20-node HEX mesh. ....	35
Figure 18. ANSYS temperature results for NBK7 window. ....	36
Figure 19. Baseline room temperature interferometry of BK7 test window. ....	38
Figure 20. Interferometry of BK7 test window under 3W load for 90 minutes. ....	39
Figure 21: Predicted steady thermal states for BK7 (left) and Fused Silica (right) lenses. Contours are 35, 45, 55, 65, 75, 85°C. ....	40
Figure 22: Predicted displacements for BK7 (left) and Fused Silica (right) lenses. Contours are 2, 4, 6, 8, and 10 microns. ....	41
Figure 23: Z-displacements (microns) as color contours on a lens-shape with exaggerated distortion.....	42
Figure 24: Z-direction “swelling” on x=0 plane comparison of ANSYS [15] [16] and Aria Predictions.....	43

## TABLES

Table 1. Shape functions for a HEX20 element. ....	15
Table 2. HEX20 shape functions on the $\zeta = +1$ surface. ....	15
Table 3. User switches for FE ray-trace analyses. ....	24
Table 4. Material properties for selected test articles. ....	27
Table 5. Labview System Calibration. ....	30
Table 6. Test Uncertainty .....	30
Table 7. Comparison of Thermocouple and IR Temperatures, NBK7 Back Side. ....	31
Table 8. Comparison of Thermocouple and IR Temperatures, NBK7 Front Side. ....	32
Table 9. Comparison of Thermocouple and IR Temperatures, FS back side. ....	33
Table 10. Comparison of Thermocouple and IR Temperatures, FS front side. ....	34
Table 11. ANSYS Model Results and Temperature Data, NBK7 Window. ....	36
Table 12. ANSYS Model Results and Temperature Data, FS Window. ....	37
Table 13: Properties used in analysis. ....	39
Table 14: Model conditions used in analysis. ....	40

## NOMENCLATURE

DOE	Department of Energy
SNL	Sandia National Laboratories
FE	Finite Element
FEM	Finite Element Model
FEA	Finite Element Analysis
LDRD	Laboratory Directed Research and Development project
TOP	Thermo-Optic Propagation
API	Application Programming Interface
UDS	Zemax User-Defined Surface
DLL	Dynamic-Link Library
MFC	Microsoft Foundation Class library

# 1. INTRODUCTION

Due to the rising complexity of space-borne optical systems, on-orbit performance requirements must increasingly be verified by analysis with validation against a subset of data collected during ground-based testing. These systems may experience dynamic thermal loads by virtue of their orbital geometries, where time-varying sun illumination angles and earth-eclipse subject them to extreme heat and cold. Since thermal loads degrade performance of an optical system by misaligning, distorting, and altering the optical properties of components, detailed thermo-optical analyses are required to complete the requirements verification process.

Current methods for performing thermo-optic analyses are extremely laborious and predominantly compartmentalized according to discipline: structural, thermal & fluid mechanics, and optical propagation analysis. While the structural and thermal engineering disciplines have increasingly gravitated toward finite-element (FE) theory to provide tractable numerical methods for modeling complex phenomena of coupled physics, optical engineering mostly relies upon ray-tracing (geometrical optics) and Fourier analysis (wave optics) to model system performance. The classical method of reducing FE data for final optical analysis involves mapping into a form compatible with the chosen tools, e.g. least-squares fit of a Zernike polynomial equation to the FE data. This mapping is not a true multi-physics implementation and yields only an approximate solution at best. It is susceptible to error and cannot account for effects of field-angle and compounded errors of upstream components in a single instantiation of the mapped approximation.

This LDRD project constructs and demonstrates a new theoretical basis for tracing optical rays directly within a finite-element (FE) volume. The classical ray-trajectory equations are cast into the local element coordinate frame and the full finite-element interpolation is used to determine the instantaneous refractive index gradient for the transformed ray-path integral equation. The FE methodology (FEM) is also used to interpolate local surface deformations and the surface normal vector for computing the refraction angle when launching rays into the volume, and again when rays exit the medium. Resulting ray trajectories are compared against closed-form gradient index methods using the Matlab™ modeling tool.

A software architecture is also developed for implementing the algorithms in the Zemax™ commercial ray-trace application. The architecture is constructed using the Zemax User-Defined Surface API, for direct implementation of FE ray-tracing within the Zemax environment. Software hooks are provided into the Zemax menu and dialog structure for custom FE analyses and development comparisons.

Finally, a controlled thermo-optic environment is assembled in the laboratory for hardware validation of the thermo-optic modeling capabilities. Surface and optical transmission data is collected for fused-silica and BK7 window substrates under various thermal loads. Thermal data is also measured and compared to model predictions under specified heat loads.



## **2. THE THERMO-OPTIC ANALYSIS PROCESS**

### **2.1 Optical System Overview**

A thorough discussion of optical system design methodologies is beyond the scope of this paper. However, a short treatise on the subject bears merit in the context of defining the overlapping roles of optical design and analysis, mechanical design, and structural analysis. Also, for systems expected to operate in extreme thermal environments, heat-transfer and thermal analyses must be included to ensure a system meets its performance requirements.

At the most basic level, an optical system is comprised of one or more optical components (lenses, windows, or mirrors) held in relative alignment by a housing or mechanical metering structure. A system may include an integrated optical sensor, or allow the collected optical energy to exit through an aperture. Each of these subcomponents is comprised of physical materials (glass, metal, plastic, etc.) that inherently possess certain properties (index of refraction, absorption coefficient, coefficient of thermal expansion, elasticity, thermal conductivity, etc.). Moreover, when subject to varying environmental parameters (temperature, pressure, force loads, etc.), the components of a system react to these changes in a manner consistent with their material properties. The components are coupled by virtue of the system design, and component changes due to environmental factors impact the overall optical performance.

Optical systems designed for the space environment are subject to extreme changes in environmental parameters. These systems are integrated and tested in laboratories under gravitational load at 1 atm pressure, and subsequently launched into orbital geometries with zero-gravity and vacuum pressure. On station, the systems may (depending on orbitology) experience both full sun illumination (without benefit of atmospheric filtering) and full earth shadow, creating time-varying and extreme thermal conditions. Since many of these parameters cannot be duplicated in the laboratory environment, verification of performance requirements must be achieved through modeling and analysis.

### **2.2 Classical Thermo-Optic Analysis**

The classical approach to thermo-optical analysis involves sequential development of expert domain models for thermal and structural analyses, with results cycled back to the optical designer for reintroduction into the original prescription file and final performance assessment. This approach is necessitated by the complexities of the required analyses and the maturity of the codes developed by and for each expert domain.

The fields of structural and thermal analysis have increasingly gravitated toward finite-element models to provide tractable numerical methods for modeling complex phenomena of coupled physics. A number of commercial tools are available to generate the discretized models and perform the specialized analyses. However, optical analysis predominantly relies on ray-tracing or Fourier wave propagation techniques. Inclusion of FE results in an optical analysis is not easily accomplished with these tools. Methods for mapping the results to a standard polynomial

equation are given in [1]. For transmissive optical elements, the compound effects of thermal distortion and the thermal dependencies of refractive index, render this mapping valid only at the specific wavelengths and field-angles for which they are derived. Broad-spectrum and broad-field analyses require the mapping to be performed many times at discrete values. This is a laborious, inflexible, and costly process, which typically relegates thermo-optic analysis to the post-design and integration phases of system development. At this late stage of design, any flaws exposed by the thermal analysis will be extremely costly to repair.

## 2.3 Finite-Element Ray Trace Analysis

By developing a capability to perform ray-tracing directly within a FE volume, these limitations are largely mitigated. FE data can be readily imported into the ray-trace application and optical parameters such as field-angle and wavelength can be varied at will, enabling rapid analyses over broad parameter sweeps and reducing costs. This capability allows thermo-optic analyses to be performed earlier in the system design cycle, enabling a higher level of optimization of the design.

While basic methodologies for finite-element ray-tracing have been explored for a number of years [2] [3] [4], the outlined techniques do not fully integrate the finite-element methods. Richerzhagen [2] over-simplified the approach by assuming a uniform FE discretization and homogeneous index within each sub-element. Epstein, *et al*, [3] limited the ray-trajectories and gradient geometries to specific cases with special application in the field of seismic wave propagation. And, Gatej, *et al*, ignored the FE connectivity and interpolation methods in favor of a nodal scattered data approximation technique for determining refractive index along the ray trajectory, relegating the ray-trace portion to an intermediate step with Zernike mapping for the final optical analysis (similar to the classical thermo-optic analysis process).

The primary contribution of this LDRD effort is the development of a finite-element ray-trace theory that fully exploits the rich methodologies of FE theory for interpolation and gradient calculations within an element volume defined by a collection of discrete nodes. The theory outlined in [5], enables ray-tracing directly within a finite-element volume and has broad application in the fields of optics and seismic wave propagation. The approach allows data files from the thermal and structural FE models to be imported into the optical ray-trace application and multi-spectral, multi-field analyses to be performed with no need for intermediate processing. The ease and reduced cost allow thermo-optic analyses to be performed earlier in the design cycle for optical systems, reducing risk and enabling a higher degree of design optimization.



### 3. RAY TRACING IN THE FINITE ELEMENT DOMAIN

#### 3.1 Ray-Trace Equations in the Finite Element Domain

The differential ray equation is given in [6], as

$$\frac{d}{ds} \left[ n(\mathbf{r}) \frac{d\mathbf{r}}{ds} \right] = \nabla n(\mathbf{r}), \quad (1)$$

where  $\mathbf{r}$  is the position vector of a point on the ray,  $n(\mathbf{r})$  is the refractive index of the medium at position  $\mathbf{r}$ , and  $ds$  is a differential portion of the arc length  $s$  along the ray. Using the change of

variable  $t \equiv \int \frac{ds}{n}$  and  $dt \equiv \frac{1}{n} ds$ , proposed in [7] and the techniques described in [5] (included in [Appendix A](#)), this equation can be expanded in Cartesian space and cast into a pseudo-Newtonian mechanics form to yield,

$$\begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \\ \frac{d^2z}{dt^2} \end{bmatrix} = \frac{1}{2} \nabla n^2 = \frac{1}{2} \begin{bmatrix} \frac{\partial n^2}{\partial x} \\ \frac{\partial n^2}{\partial y} \\ \frac{\partial n^2}{\partial z} \end{bmatrix}. \quad (2)$$

In this equation, the gradient of the index-squared acts as a "forcing function" which imparts a "pseudo-acceleration" of the ray as it travels through the medium. Integration of equation (2) over the independent variable  $t$ , yields the trajectory of the ray in  $(x, y, z)$  physical Cartesian coordinate space.

In [5], the transformation of equation (2) from  $(x, y, z)$  physical coordinates to the  $(\xi, \eta, \zeta)$  local coordinate frame of the finite-element is derived. For the FE local frame, the ray equation becomes,

$$\begin{bmatrix} \frac{d^2\xi}{dt^2} \\ \frac{d^2\eta}{dt^2} \\ \frac{d^2\zeta}{dt^2} \end{bmatrix} = \mathbf{J}^{-T} \left\{ \frac{1}{2} \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial n^2}{\partial \xi} \\ \frac{\partial n^2}{\partial \eta} \\ \frac{\partial n^2}{\partial \zeta} \end{bmatrix} - \frac{d\mathbf{J}^T}{dt} \begin{bmatrix} \frac{d\xi}{dt} \\ \frac{d\eta}{dt} \\ \frac{d\zeta}{dt} \end{bmatrix} \right\}, \quad (3)$$

where  $\mathbf{J}$  is the Jacobian matrix of the element at position  $(\xi, \eta, \zeta)$ ,  $\mathbf{J}^T$  denotes its transpose,  $\mathbf{J}^{-1}$  is the matrix inverse, and  $\mathbf{J}^{-T}$  is the inverse of the transpose matrix. It is interesting to note the right hand side of equation (3) carries two forcing function terms, while that of equation (2) carries only a single term. One of the terms in equation (3) is the analogue to the gradient of the index-squared, composed in the local FE  $(\xi, \eta, \zeta)$  domain. As explained in [5], the second term resolves the curvilinear mapping of the local FE domain to the physical coordinate space.

### 3.2 Ray-Trace Algorithm

A sequential algorithm for tracing a given ray through a finite-element volume is also described in [5]. The basic algorithm entails the following steps:

1. Find the intersection of the given ray with the surface.
2. Determine the finite-element that encompasses that intercept point.
3. Map the intercept point into the local  $(\xi, \eta, \zeta)$  frame of the intercept element.
4. Refract the ray into the element at the intercept point.
5. Cast the refracted ray into the curvilinear space of the local element frame.
6. Trace the ray through the element until it exits some surface (defined by a crossing of the  $\pm 1.0$  point in any local axis).
7. Determine if the ray transitions to a neighboring element and perform the element transition transformation, if required.
8. Repeat steps 6 and 7, until the ray exits the FE volume.
9. Refract the ray into adjacent medium upon exit from the volume.

While this algorithm appears rudimentary and simplistic, the complexity and importance of steps one through three cannot be over emphasized. Accurate determination of the ray incidence point and its mapping into the local  $(\xi, \eta, \zeta)$  coordinates of the incident element is a non-trivial task that strongly impacts the accuracy of the ray trace. A gradient-descent cost-minimization approach to determining the incidence point in  $(\xi, \eta, \zeta)$  space is proposed in Appendix B of [5]. However, during implementation of this approach a fundamental degenerate geometrical error was discovered in that formulation.

### 3.3 FE optical surface interpolation error

A typical optical surface is generally characterized by a closed-form conic-equation of revolution, with sag  $z$  given by

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}}, \quad (4)$$

where  $c$  is the curvature, defined as the reciprocal of the radius of curvature at the center (vertex) of the optical surface,  $k$  is the conic-constant, and  $r$  is the radial distance from the vertex. This closed-form equation is continuous, providing one-to-one mapping of the sag at any given point  $(x, y)$  lying on the surface. The intersection of any given ray with the surface defined by equation (4) is readily determined through simple geometry. Most ray-tracing applications employ some form of this equation in their respective ray-trace engines.

When discretizing the surface of equation (4) for finite-element analysis, a number of nodes are defined on the surface which must also satisfy the conditions of the conic equation. A connectivity mapping is then implemented to define elements with surfaces that lie on the conic surface. In the fundamental approach of [5], FE methods provide a means to interpolate certain parameters between the nodes of the defined element using equations of the form:

$$p(\xi, \eta, \zeta) = \sum_{i=1}^I N_i p_i \quad (5)$$

where  $p$  is the parameter of interest,  $p_i$  is the value of the parameter at the  $i^{\text{th}}$  node,  $N_i$  is the  $i^{\text{th}}$  shape function, and  $I$  corresponds to the number of nodes contained in the element (e.g.  $I = 20$  for a HEX20 element). The shape functions are dependent on the number of nodes contained in the element and the local coordinate position  $(\xi, \eta, \zeta)$ . For a HEX20 element, the shape functions are given in Table 1, and provide 2<sup>nd</sup>-order parameter interpolation throughout the volume between the nodes.

**Table 1. Shape functions for a HEX20 element.**

Shape Function
$N_1 = -\frac{1}{8}(1 - \xi)(1 - \eta)(1 - \zeta)(2 + \xi + \eta + \zeta)$
$N_2 = -\frac{1}{8}(1 + \xi)(1 - \eta)(1 - \zeta)(2 - \xi + \eta + \zeta)$
$N_3 = -\frac{1}{8}(1 + \xi)(1 + \eta)(1 - \zeta)(2 - \xi - \eta + \zeta)$
$N_4 = -\frac{1}{8}(1 - \xi)(1 + \eta)(1 - \zeta)(2 + \xi - \eta + \zeta)$
$N_5 = -\frac{1}{8}(1 - \xi)(1 - \eta)(1 + \zeta)(2 + \xi + \eta - \zeta)$
$N_6 = -\frac{1}{8}(1 + \xi)(1 - \eta)(1 + \zeta)(2 - \xi + \eta - \zeta)$
$N_7 = -\frac{1}{8}(1 + \xi)(1 + \eta)(1 + \zeta)(2 - \xi - \eta - \zeta)$
$N_8 = -\frac{1}{8}(1 - \xi)(1 + \eta)(1 + \zeta)(2 + \xi - \eta - \zeta)$
$N_9 = \frac{1}{4}(1 - \xi)(1 + \xi)(1 - \eta)(1 - \zeta)$
$N_{10} = \frac{1}{4}(1 + \xi)(1 - \eta)(1 + \eta)(1 - \zeta)$
$N_{11} = \frac{1}{4}(1 + \xi)(1 - \xi)(1 + \eta)(1 - \zeta)$
$N_{12} = \frac{1}{4}(1 - \xi)(1 - \eta)(1 + \eta)(1 - \zeta)$
$N_{13} = \frac{1}{4}(1 - \xi)(1 + \xi)(1 - \eta)(1 + \zeta)$
$N_{14} = \frac{1}{4}(1 + \xi)(1 - \eta)(1 + \eta)(1 + \zeta)$
$N_{15} = \frac{1}{4}(1 - \xi)(1 + \xi)(1 + \eta)(1 + \zeta)$
$N_{16} = \frac{1}{4}(1 - \xi)(1 - \eta)(1 + \eta)(1 + \zeta)$
$N_{17} = \frac{1}{4}(1 - \xi)(1 - \eta)(1 - \zeta)(1 + \zeta)$

$N_{18} = \frac{1}{4}(1 + \xi)(1 - \eta)(1 - \zeta)(1 + \zeta)$
$N_{19} = \frac{1}{4}(1 + \xi)(1 + \eta)(1 - \zeta)(1 + \zeta)$
$N_{20} = \frac{1}{4}(1 - \xi)(1 + \eta)(1 - \zeta)(1 + \zeta)$

In the FE construct, the local coordinates are bounded in the range  $[-1;1]$ . Therefore, at the optical surface, one of the coordinate parameters must be  $\pm 1$ . Setting  $\zeta = 1$  in the shape function equations of Table 1, yields:

**Table 2. HEX20 shape functions on the  $\zeta = +1$  surface.**

Shape Function
$N_1 = 0$
$N_2 = 0$
$N_3 = 0$
$N_4 = 0$
$N_5 = -\frac{1}{4}(1 - \xi)(1 - \eta)(1 + \xi + \eta)$
$N_6 = -\frac{1}{4}(1 + \xi)(1 - \eta)(1 - \xi + \eta)$
$N_7 = -\frac{1}{4}(1 + \xi)(1 + \eta)(1 - \xi - \eta)$
$N_8 = -\frac{1}{4}(1 - \xi)(1 + \eta)(1 + \xi - \eta)$
$N_9 = 0$
$N_{10} = 0$
$N_{11} = 0$
$N_{12} = 0$
$N_{13} = \frac{1}{2}(1 - \xi)(1 + \xi)(1 - \eta)$
$N_{14} = \frac{1}{2}(1 + \xi)(1 - \eta)(1 + \eta)$
$N_{15} = \frac{1}{2}(1 - \xi)(1 + \xi)(1 + \eta)$
$N_{16} = \frac{1}{2}(1 - \xi)(1 - \eta)(1 + \eta)$
$N_{17} = 0$
$N_{18} = 0$
$N_{19} = 0$
$N_{20} = 0$

The eight non-zero functions are identical to the shape functions of a QUAD8 plate element, which is consistent with interpolation on any surface of a HEX20 solid element. In expanded

application, the QUAD8 shape functions provide second-order interpolation of the nodal parameters in the form:

$$p = a_1 + a_2x + a_3y + a_4xy + a_5x^2 + a_6y^2 + a_7x^2y + a_8xy^2. \quad (6)$$

Using this equation to interpolate the sag  $z$ , it can be shown that except for special cases of the conic-constant  $k$ , equation (6) and equation (4) cannot generally be made to yield identical results for any given values  $a_1 - a_8$ . Therefore, the FE interpolation shape functions are generally inadequate for describing a conic-equation of revolution on the optical surface.

As an example, consider a spherical optical surface ( $c \neq 0, k = 0$ ) with a QUAD8 (or, HEX20) finite element located at the vertex and the surface nodes residing at equal radial distances from the vertex, such that the  $z$ -coordinate values are identical for all eight nodes on the surface. For this case, all eight nodes lie in a plane, parallel to the  $XY$  plane. Any interpolation scheme based only on the eight nodal coordinate values will yield a point which also lies in the plane. The error from the true surface can be dramatic when considering optical elements with a steep radius of curvature. By extension, the FE interpolation methods are generally inadequate for finding the intersection of a given ray with the true optical surface. Indeed, this is a shortfall of the FE interpolation method as applied to the ray-trace formulation.

To overcome this shortfall, it is required to allow the local FE coordinates to lie outside the  $[-1;1]$  bound when interpolating points on the surface of the optic. Then, the error metric proposed in [5] must be redefined for determining the ray-intersection point on the surface.

### 3.4 Ray intercept calculation

To maintain consistency between the FE interpolation and the surface of revolution equations, a new method for calculating the ray intercept is required. The approach derived for this LDRD is an extension of the classic ray-direction method. In a homogeneous (ambient) medium, rays maintain a linear trajectory. Therefore, given a specific ray defined by its direction cosines ( $\alpha, \beta, \gamma$ ) and a single physical point ( $x_0, y_0, z_0$ ) through which it passes, any other point ( $x_1, y_1, z_1$ ) lying on the ray trajectory may be determined by

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + t \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}, \quad (7)$$

where  $t$  is the distance between the two points. Note that  $[\alpha, \beta, \gamma]$  is a unit vector.

If ( $x_1, y_1, z_1$ ) also lies on an optical surface, the values must simultaneously satisfy the surface equation, e.g. equation (4). Explicitly, this yields,

$$z_1 = \frac{c(x_1^2 + y_1^2)}{1 + \sqrt{1 - (1 + k)c^2(x_1^2 + y_1^2)}}. \quad (8)$$

In the classic formulation, equations (7) and (8) are combined and rearranged to provide a closed-form solution for  $t$ , which can be substituted back into equation (7) to yield the intercept point.

### 3.4.1 Intercept of the nominal surface

To find the FE local coordinates of the intercept point, it is necessary to define a consistent set of equations between the FE shape-function interpolation and the surface equation. For the nominal (load-free) finite-element lying on the surface of the optic, it is easy to see from equation (5) that,

$\bar{x} = \sum_{i=1}^I N_i x_i$  ;  $\bar{y} = \sum_{i=1}^I N_i y_i$  ; and  $\bar{z} = \sum_{i=1}^I N_i z_i$  . But if the point defines the intersection of a ray with an optical surface, it must also satisfy equations (7) and (8).

Applying the constraint of equation (8), yields

$$\bar{z} = \sum_{i=1}^I N_i z_i = \frac{c[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}{1 + \sqrt{1 - (1+k)c^2[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}}$$

And, combining these constraints leads to the vector error metrics,

$$\begin{aligned} \mathbf{e}_1 &= \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix}; \\ &= \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + t \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} - \begin{bmatrix} \sum N_i x_i \\ \sum N_i y_i \\ \frac{c[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}{1 + \sqrt{1 - (1+k)c^2[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}} \end{bmatrix}, \\ \mathbf{e}_2 &= \sum N_i z_i - \frac{c[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}{1 + \sqrt{1 - (1+k)c^2[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}}, \end{aligned} \quad (9)$$

and,

which can be assembled into a total error vector,

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}.$$

The error cost metric is defined as the inner product (dot product, or  $H_2$  norm) of the error. A gradient descent algorithm may be employed to minimize this cost, as described in [5].

Equations (9) capture the constraint space of the full 3D local coordinate frame of a solid element. When used to generate the cost value for gradient minimization, the search parameters are  $(\xi, \eta, \zeta, t)$ . The result yields the proper local coordinates  $(\xi, \eta, \zeta)$  of the ray intercept point on the surface of the incident element, allowing for absolute local coordinate values in excess of 1.0

to enforce the equivalence of the surface equation and the FE interpolation shape-functions. It also yields the ray path length  $t$ , allowing the physical coordinates to be derived from equation (7).

However before this minimization can be applied, the true incident element must first be located. Reference [5] outlines an iterative approach for finding the incident element, using only the “glued-on” QUAD8 plate element of the optical surface. In this approach, the local coordinates of a candidate plate-element are determined using only the first three components of equation (9). Since plate elements carry only a 2D local coordinate frame, the result must be bounded by the  $\pm 1.0$  constraint of the element. If the result lies outside this bound, the neighboring elements are queried in an iterative fashion until the properly bounded plate element is located, or the ray is determined to have missed the optical surface. When the proper element is discovered, the full equation (9) is applied to the underlying solid element to derive the 3D local coordinates of the intercept point.

### 3.4.2 Intercept of the distorted/displaced surface

Under exogenous load, the nodes comprising the optical surface are displaced. These displacements are captured as separate nodal parameters, which are interpolated in the common approach of equation (5). For displaced nodes, the coupling equations between the FE shape-functions and the surface equation must be modified. Moreover, the modification must be consistent when applied to both rigid-body motions of the optic and compliant deformations of the surfaces.

The coupling system proposed in this LDRD assumes linear summation of displacement with nominal position of the surface. When the surface equation is the standard conic of revolution, this is represented by,

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} \sum N_i x_i \\ \sum N_i y_i \\ \frac{c[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}{1 + \sqrt{1 - (1+k)c^2[(\sum N_i x_i)^2 + (\sum N_i y_i)^2]}} \end{bmatrix} + \begin{bmatrix} \sum N_i \delta x_i \\ \sum N_i \delta y_i \\ \sum N_i \delta z_i \end{bmatrix}.$$

This form can replace the terms for  $e_1$  of equation (9), when finding the ray intercept of optical components under load. Considering rigid-body displacements, the  $\delta x_i$ ,  $\delta y_i$ , and  $\delta z_i$  are defined as constant for all  $i$ . Inclusion of the sag function for  $z$ , based on the nominal interpolation for  $x$  and  $y$  (without displacements), ensures consistency in the coupling.

## 4. IMPLEMENTATION IN THE ZEMAX USER-DEFINED SURFACE

### 4.1 The User-Defined Surface (UDS) API

The Zemax ray-trace application provides a C-code application programming interface (API) for developing custom surface types [8]. The API uses two function argument structures to pass data to-and-from the user's code in a *Client-Server* architecture. It is similar to a callback function, applying a master *switch-case* structure to service requests made by Zemax when executing specific ray-tracing tasks. The reader is directed to the Zemax manual for further description of the API.

It is important to note that the Zemax UDS can only be used in sequential ray-trace mode. In this mode, Zemax requires rays to be traced sequentially from surface-to-surface until they strike the final surface of the system, or they miss the next surface in the optical train. A ray cannot strike the same surface twice. If a ray misses a surface, Zemax stops tracing that ray. Also in this mode, the surfaces define the interfaces between optical media (e.g. air-to-glass, or glass-to-air transitions). Zemax assumes the optical media between surfaces to be homogeneous, and performs the calculation to trace a ray from one surface to the next.<sup>1</sup> When a ray is incident upon a UDS, Zemax passes ray information to the UDS code and requests information about the refraction of the ray at the surface, including the point-of-incidence, the surface slope at that point, and the direction of the refracted ray. Zemax then traces the ray through the medium to the next sequential surface.

Conversely the FE ray-trace is implemented through adjacent FE *volumes*, not surfaces. Therefore, when implementing the FE ray-trace within a UDS, it is necessary to provide code-traps for special cases, to enforce proper sequential operations within the API. Further, the FE volume is defined throughout the region between two sequential surfaces. Significant attention must be applied to managing the volumetric aspects within the sequential-surface API provided by Zemax.

The Thermo-Optic Propagation (TOP) code developed under this LDRD uses an optical *singlet* data structure to implement the FE ray-trace construct. An optical singlet has a front and back surface, defined by a radially-symmetric equation of revolution, and is comprised of an optical medium - in this case an FE volume. The FE volume is comprised of an optical material that has been discretized into nodes with a nodal connectivity map that defines the finite-elements. The majority of the FE data is supplied by FE analysis and is read into custom objects and structures within the UDS as a text file. The TOP code architecture makes extensive use of the Microsoft Foundation Class (MFC) libraries to manage the optical singlet data using the MFC Document/View architecture.

---

<sup>1</sup> Special surface-types exist in the Zemax library to introduce gradient-index (inhomogeneous) media into a lens file. Only a handful of such surfaces exist, and each implements a special closed-form equation describing the gradient to use during the ray-trace between surfaces.



Much effort was expended in the TOPCode development to create a common-look, common-feel to the Zemax intrinsic surface library. The TOPCode UDS is inserted as a surface into a lens prescription file in the same way as any other surface type. The first instance of the UDS requests the user to identify the FE data file for import.<sup>2</sup> The UDS will then inform the user/designer that a second TOPCode UDS surface is required to sequentially follow the first. The sequential pair of TOPCode surfaces define the front and back surfaces of the FE singlet. The TOPCode UDS manages this pair of surfaces for ray-trace and analysis.

Multiple FE singlets may be inserted into any lens prescription file, using pairs of TOPCode surfaces to define each optical component. The TOPCode UDS manages the FE data associated with each singlet, to provide high fidelity ray-tracing and analysis.

## 4.2 Zemax ray-trace requests

When performing sequential ray-tracing through a lens system, Zemax uses the optical-surface information listed in the Lens Data Editor (*Surface Type*, *Radius* of curvature, *Glass* type, etc.) to calculate the refraction at the surface. It then propagates the ray over the *Thickness* of the surface, and repeats the process at each sequential interface. When a UDS is encountered in the sequence, Zemax passes the incident ray parameters to the user's custom code, and issues a flag requesting the ray refraction information. The UDS code must return this information in the argument structure defined by the API. Zemax then uses this information to propagate the ray over the thickness, using the optical properties of the listed glass-type

The TOPCode UDS must operate within this API construct. When Zemax requests refraction information for a ray incident on the front surface of the FE singlet, the UDS must:

1. Determine the incidence point of the ray with the (possibly deformed/displaced) front surface.
2. Find the finite-element containing the incidence point.
3. Interpolate the index of refraction and surface normal at the incidence point.
4. Refract the ray.
5. Return the refraction information to Zemax in the required structure.

Zemax will then trace the ray through the listed thickness of the surface, until it strikes the back surface of the FE singlet. Since the back surface is also a UDS, Zemax will issue the same ray-trace call to the UDS using the propagated ray definition parameters for this surface of the FE singlet.

The TOPCode UDS must recognize the Zemax request is for the back surface of the singlet. Since Zemax assumed a homogeneous medium for calculating the incoming Zemax ray information, the UDS must:

1. Find the intersection of the ray with the front surface.
2. Find the finite-element containing the incidence point.
3. Cast the ray trajectories into the local coordinate frame of the incidence element.
4. Re-trace the ray to the back surface using FE ray-trace methods in the (possibly inhomogeneous) FE volume.

---

<sup>2</sup> The FE data file must have a specific format, with a number of sections defining critical data for the ray-tracing. See [Appendix E](#) for definition of the required FE file format.

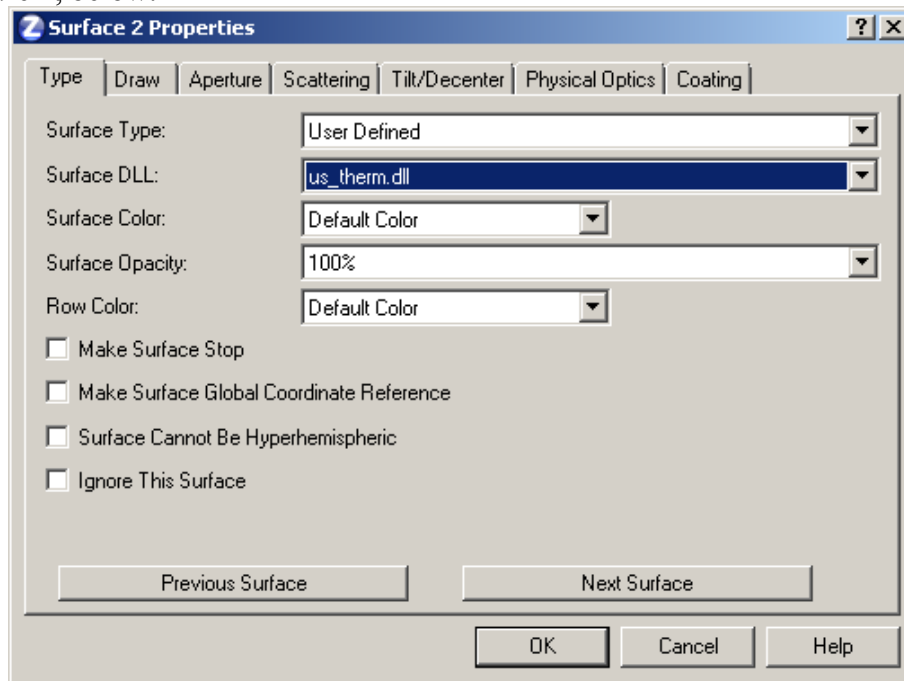
5. Re-cast the ray into physical coordinates at the back surface.
6. Interpolate the index of refraction and surface normal at the incidence point.
7. Refract the ray at the back surface.
8. Return the refraction information to Zemax in the required structure.

Whether Zemax is requesting ray information for the front or back surfaces of the FE singlet, proper determination the ray incidence point with the front surface is critical. Hence, the modifications to the ray intercept calculation outlined in section 3.4 are also critical.

## 4.3 Using the FE surface in a Zemax lens file

### 4.3.1 Modeling an FE singlet in a lens file

The FE surface is a custom user-defined surface. To use the surface type in a Zemax lens file, the user must select the *User Defined* surface type from the *Type* tab of the *Surface Properties* dialog in the Zemax Lens Data Editor. When this surface type is chosen, the *Surface DLL* edit box becomes enabled, allowing the user to select the *us\_therm.dll* from the drop-down list, as show in Figure 1, below.



**Figure 1. The Zemax Surface Properties dialog.**

When the user selects OK, a new popup dialog is generated, requesting the name of the FE data file to use for the FE volume being defined, as shown in Figure 2. This dialog allows the user to navigate to the correct FE data file. The data file must have a structure similar to that described in Appendix E.

After selecting the proper file, the warning message of Figure 3 is displayed, indicating that a second, sequential UDS is required in the lens file to complete the FE singlet definition. The user must insert a new surface in the Lens Editor, and repeat the surface-type selection to

complete the FE volume definition. Since the UDS manages the pair of FE surfaces using data contained in a single FE data file, the user is not prompted for the FE data file a second time. The final lens file listing for the FE singlet is shown in Figure 5.

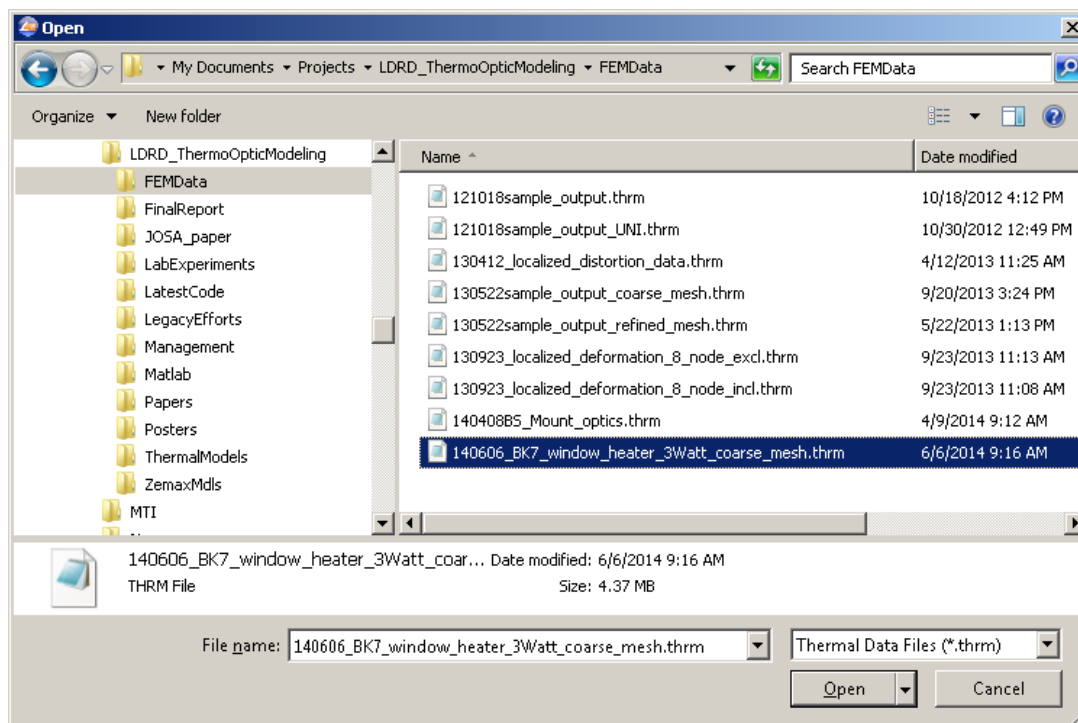


Figure 2. Open-File dialog for the FE Data File.

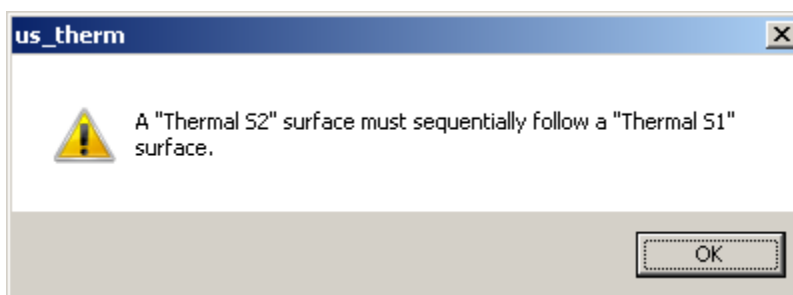
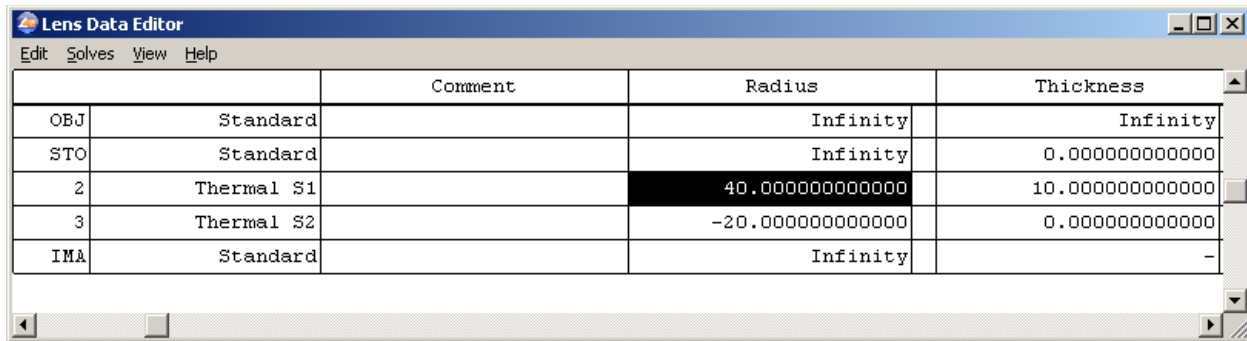


Figure 3. Dialog showing requirement for a second UDS surface.

Lens Data Editor				
Edit Solves View Help				
Surf:	Type	Comment	Radius	Thickness
OBJ	Standard		Infinity	Infinity
STO	Standard		Infinity	0.000000000000
2	Thermal S1		Infinity	0.000000000000
IMA	Standard		Infinity	-

Figure 4. Lens file listing for the first surface of the FE singlet.



	Comment	Radius	Thickness
OBJ	Standard	Infinity	Infinity
STO	Standard	Infinity	0.000000000000
2	Thermal S1	40.000000000000	10.000000000000
3	Thermal S2	-20.000000000000	0.000000000000
IMA	Standard	Infinity	-

**Figure 5. Complete lens file listing for the FE singlet, showing front and back surface.**

Additional FE singlets may be added to the lens file in a similar fashion, with each pair of surfaces requiring a separate FE data file.

#### 4.3.2 Typical use-case scenario

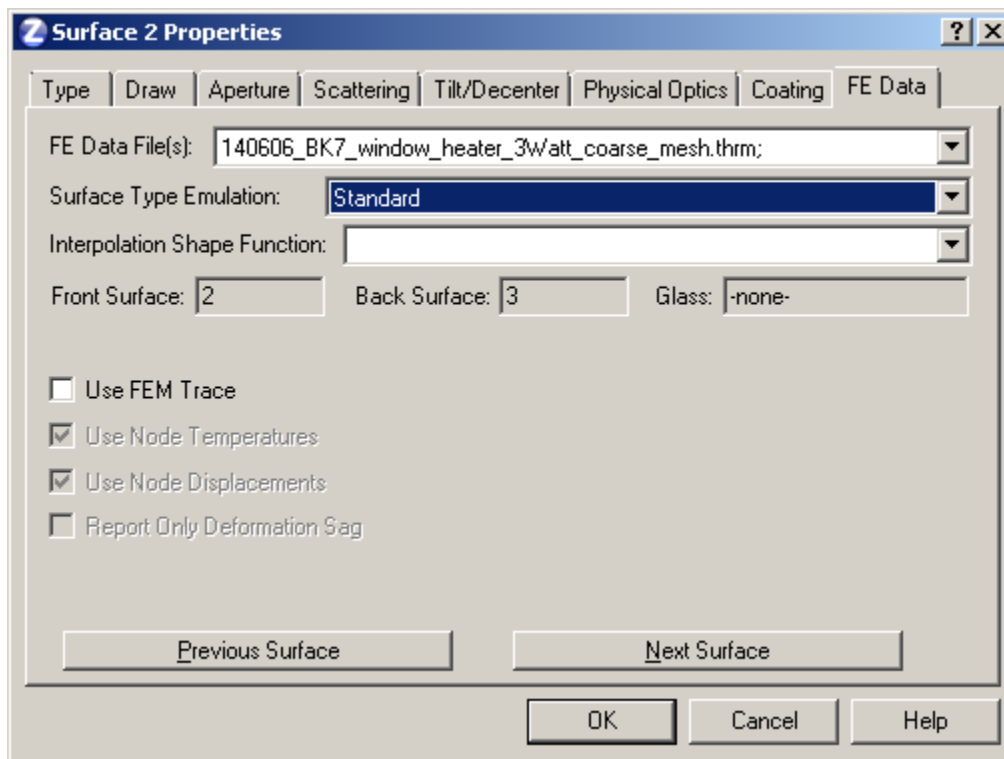
Note in Figure 5 that the user must also specify the radius of curvature and nominal thickness of the FE singlet. In the typical use-case scenario for this tool, the design flow-cycle for an optical system begins with design and optimization of the lens prescription file, using intrinsic surface types (conic surfaces of revolution) from the Zemax catalog. This optimization creates a “floating” optical design, with no supporting structure or housing.

A solid model is then exported for use by the opto-mechanical team in designing the necessary supporting structure. The support structure is designed to hold the optics in alignment within a required tolerance, under specified environmental loads. Analysis of the structural performance is accomplished using FE methods. The FE models generated during these analyses are used to export an FE data file of the form listed in Appendix E, which is delivered back to the optical designer. The optical designer then modifies the original design file, using UDS surfaces and entering the FE data files as appropriate. Final lens performance is then analyzed for the system under load.

In this scenario, the original lens optimization will determine thicknesses, radii of curvature, and other coefficients for the conic equations of revolution. The designer may choose higher-order aspheric surface types, requiring additional data. Since the exported solid model and returned FE data files do not carry this information to the precision necessary for ray-trace operations, the UDS requires this information to be maintained by the optical designer. For this implementation, the UDS maintains a common-look, common-feel to the intrinsic Zemax surface types.

To change parameters of the UDS surface and how it is accessed by the ray-trace engine, the *Surface Properties* dialog is augmented to include an additional *FE Data* tab for this surface type, as shown in Figure 6. In this custom property sheet, the user may change the specified FE data file, the Surface-Type emulator, or various other parameters of the FE model and corresponding analysis behavior. The Surface-Type emulator allows the user to include specifications of the optimized lens design that may be lost in the FE data file, including higher-order aspheric surface types. The specified surface type and coefficients entered into the Lens

Data Editor define the surface equation used by the ray-intercept method described in section 3.4.



**Figure 6. The FE Data tab of the *Surface Properties* dialog.**

This tab also provides other switches to modify certain portions of the FE ray-trace calculations when performing optical analyses. The function of these switches is outlined in Table 3, below.

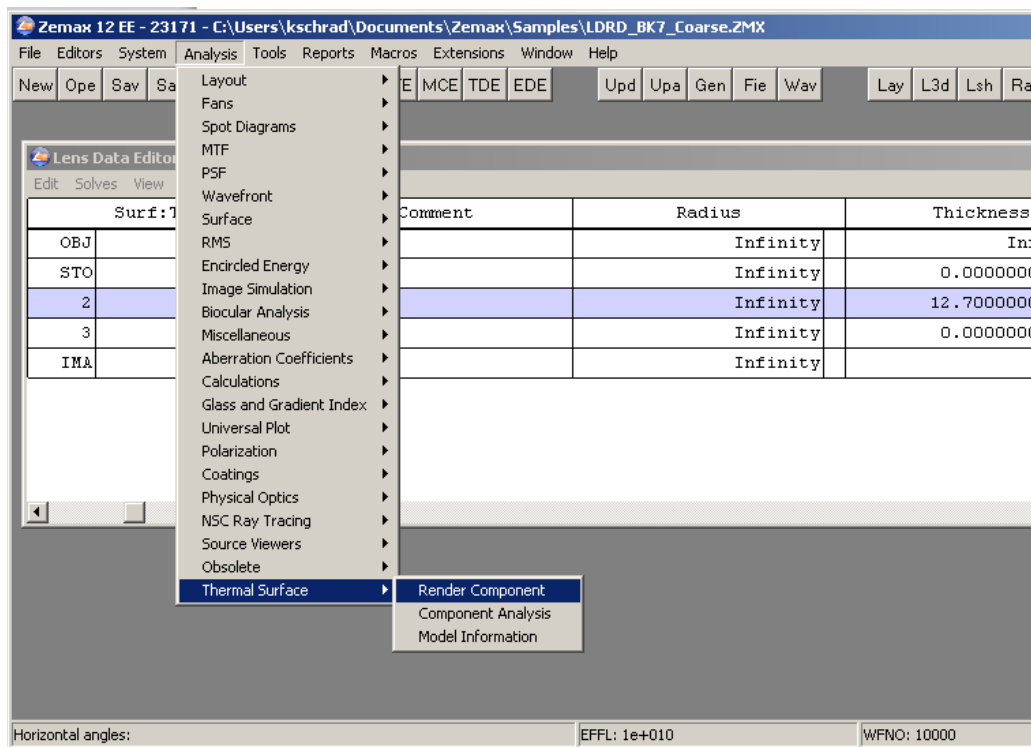
**Table 3. User switches for FE ray-trace analyses.**

Switch	State	Description
Use FEM Trace	unchecked	The UDS reverts to conic equation emulation only, and does not perform FEM ray tracing.
	checked	FE ray-trace is performed according to the other switches.
Use Node Temperature	unchecked	Homogeneous refractive index of the bulk material is used in the FE ray-trace calculation.
	checked	The UDS calculates the temperature-dependent index of refraction at each node and uses FE methods to interpolate the index gradient in the FE ray-trace calculation.
Use Node Displacements	unchecked	The nominal surface equations are used in the FE ray-trace calculation.
	checked	The UDS includes surface deformations due to thermal or structural loads in the FE ray-trace calculation
Report Only Deformation Sag	unchecked	The total sag is reported upon each Zemax request.
	checked	The UDS does not include bulk sag of the surface due to radius of curvature when responding to Zemax sag requests.

Combinations of these switches allow the analyst to validate UDS calculations against those of the intrinsic surface types, and to independently explore impact of various aspects of the thermal and structural loads with the chosen material properties. For example, if the temperature effects have a higher impact on optical performance than the surface deformations, the designer may choose to find a material with lower  $dn/dt$ , or a higher thermal conductivity to more rapidly disperse the thermal load. Or, a material with lower CTE may be required to reduce surface distortions, if that is the limiting factor in the optical performance.

### 4.3.3 Custom Analyses

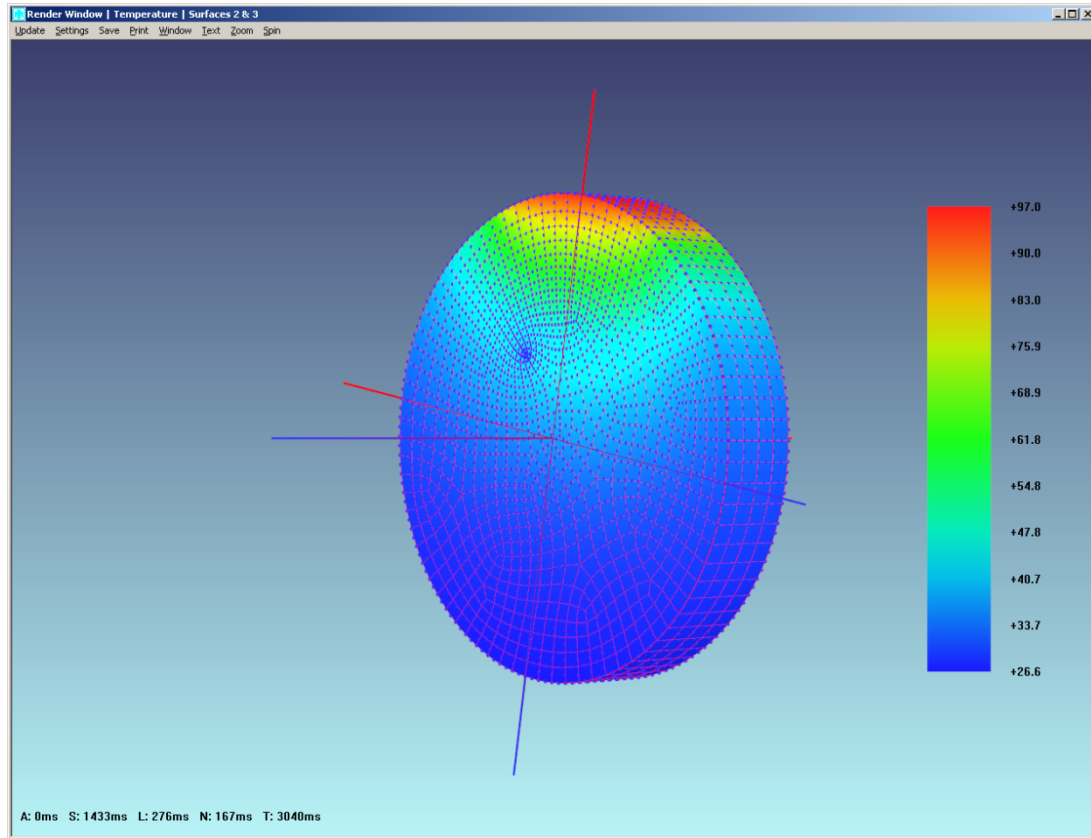
Since the UDS is a custom dynamic-link library (DLL) loaded into the Zemax address space at run-time, the MFC library provides programming functions to gain access to the Zemax root resources, including the main Zemax menu, and the Zemax message-loop in the Windows operating system. This capability allows the Zemax menu to be augmented with custom analyses when the UDS is loaded into a prescription file. This feature is demonstrated in Figure 7, where the *Thermal Surface* pop-up menu items have been appended to the bottom of the *Analysis* selection menu. The code structure for this capability is open source within the UDS software project. So, this capability can be extended for any future items that become evident as the tool is matured.



**Figure 7. Custom analyses inserted into the Zemax main menu.**

The *Render Component* analysis is a particularly useful custom capability developed within the code. This item creates a custom rendering window to visualize aspects of the finite-element singlet in its local coordinate frame. This analysis enables multiple aspects of model correlation for the coupled system. Rendering of the local FE coordinate axes assists the analyst in mating the coordinate systems within the Zemax master frame. Color shading may be attached to

specific nodal parameters, such as temperature or nodal displacement. And, rendering of the node locations and element connectivity allow correlation of localized optical effects with the FE geometry and structure. Figure 8 shows the rendering for the validation test model, where it is seen that the exported mesh geometry is highly asymmetric. This is a common artifact of auto-mesh software used in the field of FEA. Indeed, mesh optimization is a subset expert subject matter within this field. Since the UDS and FE ray-trace theory are built upon the underlying FE methods, it easily accommodates any mesh exported by the FE modeler.



**Figure 8. Rendering of the finite-element singlet in its local coordinate frame.**

#### **4.3.4 UDS Software Status**

The foundation of the software architecture for the UDS is complete and fairly mature, providing flexibility for adding new features and capabilities as required. However, the mathematical error and mitigating construct described in sections 3.3 and 3.4, were only discovered in the final days of the project. As such, the UDS is not yet ready for deployment to the broader community. Completion of the UDS will require implementation of the intercept calculations outlined in section 3.4, using other funding mechanisms.

## 5. MODEL VALIDATION AND TESTING

### 5.1 Test Description

As part of the TOP LDRD, a series of thermal tests were conducted to obtain interferometer, temperature and IR results for N-BK7 and Fused Silica (FS) windows. The windows were 0.5 inches thick and diameter of 3-inches. A summary of window properties is given in Table 3. FS has a CTE an order of magnitude lower than N-BK7 and a higher thermal conductivity.

**Table 4. Material properties for selected test articles.**

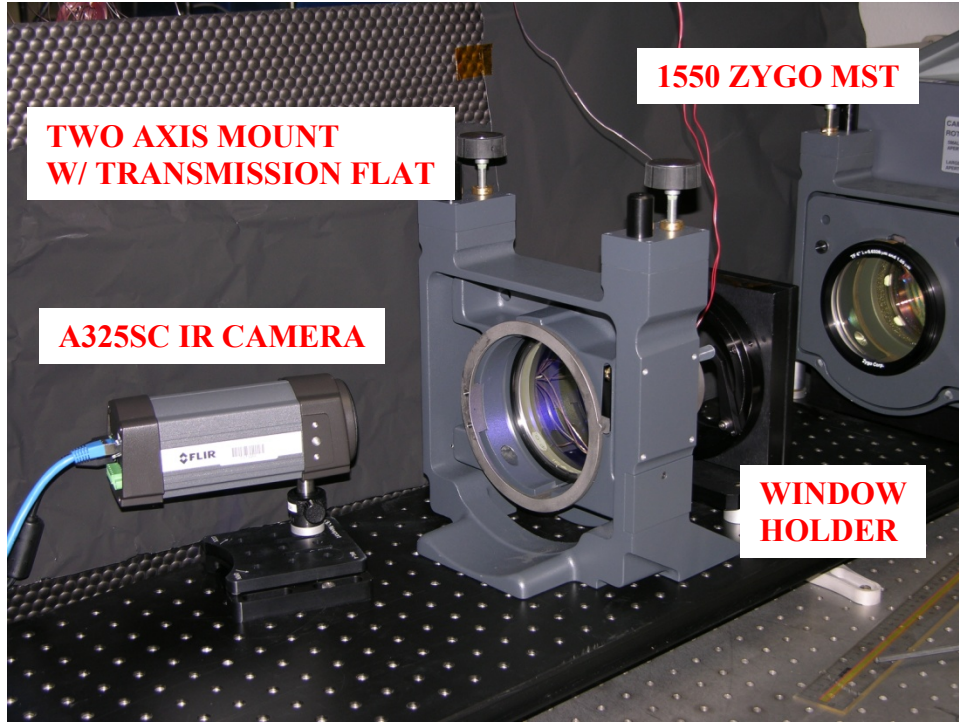
<b>Material</b>	<b>Density (g/cm<sup>3</sup>)</b>	<b>Thermal Conductivity (W/m-°K)</b>	<b>Specific Heat (J/kg-°K)</b>	<b>Coefficient of Thermal Expansion (CTE) (1/°K)</b>
N-BK7 [9]	2.51	1.114	858	$8.3 \times 10^{-6}$
Fused Silica [10]	2.2	1.38	740	$0.55 \times 10^{-6}$

The windows were held with a self-centering element holder on a 2-axis mount and supported on an optics bench. The window was heated on the outer surface with a Kapton insulated heater. Window temperatures were recorded with Type-T thermocouples adhesively attached to the window's back surface. Steady-state temperatures were recorded for heater powers of 1 W, 2 W, and 3 W, respectively. Measured temperatures were compared to infrared (IR) camera measurements and thermal analysis software tools (ANSYS, Sierra/Aria).

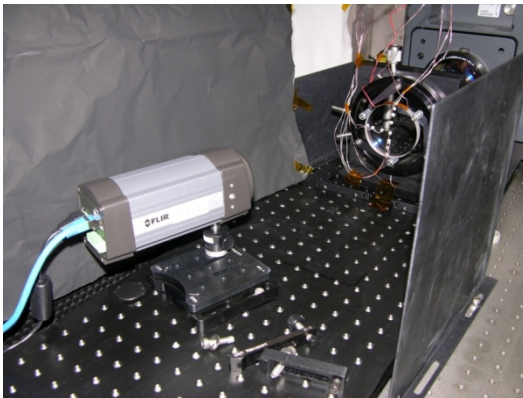
### 5.2 Interferometer Test Setup

A photograph of the four-surface interferometer test setup is shown in Figure 9. A 1550nm MST interferometer measures the optical performance of the window's front and back surfaces. A Kapton insulated heater [11] (Minco HK5576R45.0L12A) was attached to the edge of the window with an acrylic pressure sensitive adhesive (PSA), 3M966. The heater resistance tolerance is  $\pm 10\%$  with 30 AWG wire and Teflon insulated leads. The maximum heater current capacity is 3 amps and at 75°C has a maximum power density of 10 W/in<sup>2</sup> [11]. The heater width and length dimensions are 0.5 by 1.5 inches, and a maximum thickness of 0.012 inches. Heater power was provided with an external power supply. After interferometer measurements were made, the two-axis mount was removed. Thus, the interferometer was recalibrated after the mount was reinstalled. IR images were recorded with an A325sc FLIR camera located approximately eleven inches from the optic window. The camera was also tilted slightly off-axis to remove reflections from the warm camera body. During IR measurements, the camera and part of the forward portion of the setup was covered with a black coated aluminum shield to minimize background reflections.

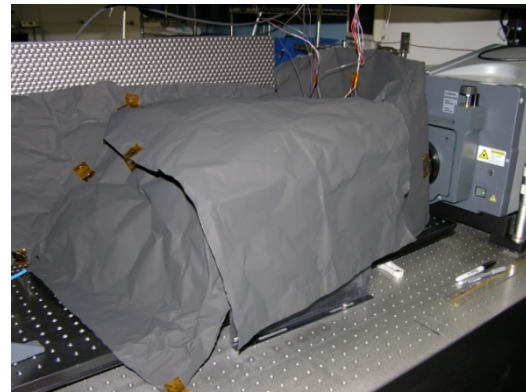




**Figure 9. Four-surface interferometer test setup.**



**(a)**

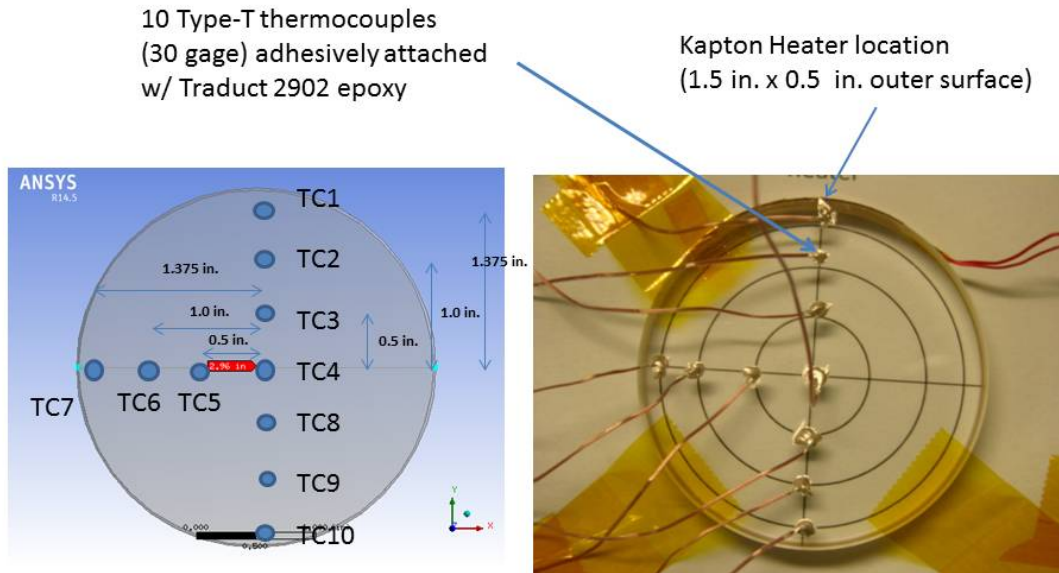


**(b)**

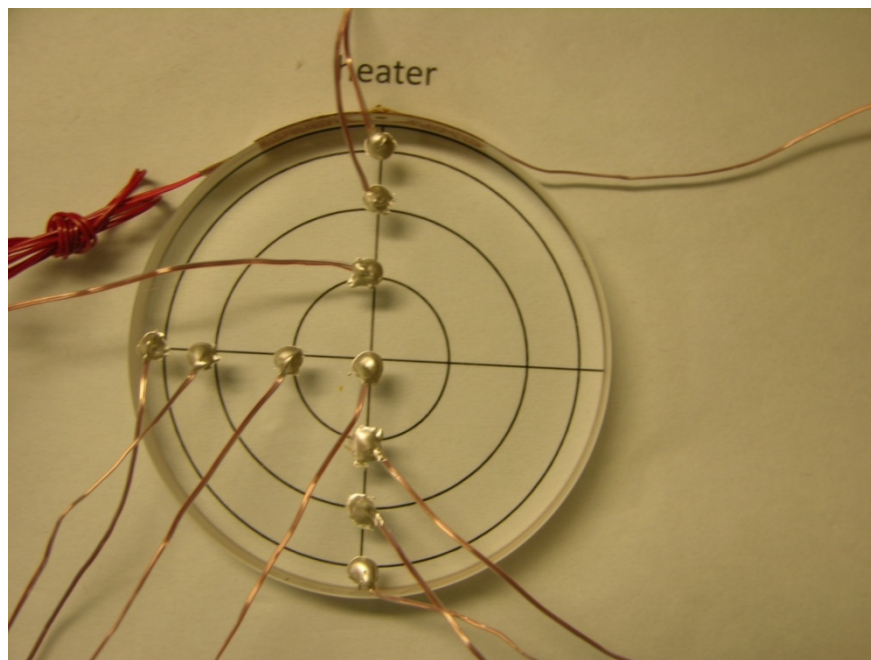
**Figure 10. Infrared camera test setup, (a) Two-axis mount removed, (b) black aluminum cover.**

Omega Type-T thermocouples (5SC-TT-T-30-180) [12] were attached to the back surface of the windows using Traduct 2902 adhesive. Figure 11 shows the locations of the thermocouples on the NBK7 window and similar locations for FS in Figure 12. It was difficult to exactly locate the thermocouples and the position uncertainly could be up to  $\pm 0.125$  inches. A thermocouple also recorded the nearby air and heater temperatures. Temperatures were recorded using a Labview data acquisition system. The Labview system was calibrated using an Omega CL123 calibrator for input temperatures ranging from 20 to 100°C. A summary of calibrated results is given in Table 4, where a gain slope error is evident in the measured values being higher at temperatures below 40°C and lower at temperatures above 40°C. Tests were performed with heater powers of

1, 2, and 3 W, respectively. Based on heater resistance tolerance, the heater power uncertainty ranged from  $1 \pm 0.1$  W to  $3 \pm 0.31$  W, respectively. A summary of test uncertainty measurements is given in Table 5.



**Figure 11. Thermocouple locations on N-BK7 window.**



**Figure 12. Thermocouple locations on FS window.**

**Table 5. Labview System Calibration.**

Calibrator Input Temp (°C )	Average Measured Temp (°C )	Average Measured STDEV (°C )
20.00	20.66	0.17
40.00	40.19	0.22
60.00	59.30	0.14
80.00	78.03	0.12
100.00	96.07	0.16

**Table 6. Test Uncertainty**

Test Parameter	Uncertainty Value
Heater power (V and I)	$\pm 0.07$ W (1 W) to $\pm 0.12$ W (3 W)
Heater Power (V and R)	$\pm 0.1$ W (1 W) to $\pm 0.31$ W (3 W)
Thermocouple Location	$\pm 0.125$ inches
Labview Thermocouple Calibration	$0.7 \pm 0.14^\circ\text{C}$
Thermocouple Calibration	$-0.3^\circ\text{C}$

## 5.3 Temperature and IR Measurements

### 5.3.1 NBK7 Optical Material

Both temperature and IR measurements were obtained for the front and back side (thermocouple side) for heater powers of 1W, 2W and 3W until steady-state conditions were obtained. IR images are shown in Figure 13, for the back surface for 1W and 3W. The gray images provide a clearer view of the surrounding hardware and thermocouple locations compared to the color plot. The infrared emissivity ( $\epsilon$ ) for the camera was set at 0.85, distance from camera to window, 0.33 meters, and air temperature,  $20^\circ\text{C}$ . The thermocouple bead temperatures are lower due to the lower emissivity value of the epoxy. The 0.85 value was obtained by comparing the thermocouple temperature to a window temperature adjacent to the bead. The IR images indicate that the support pads on the window holder did not influence the isotherms, and as expected the highest temperatures are at the top near the heater. The heater power settings provided vertical temperature differences of 28 and  $63^\circ\text{C}$ . Table 6 compares the thermocouple and IR temperatures and shows average temperature differences of  $-0.26 \pm 0.73^\circ\text{C}$  and  $-1.0 \pm 0.85^\circ\text{C}$  for 1W and 3W.

## BK7 WINDOW TC SIDE (deg. C)

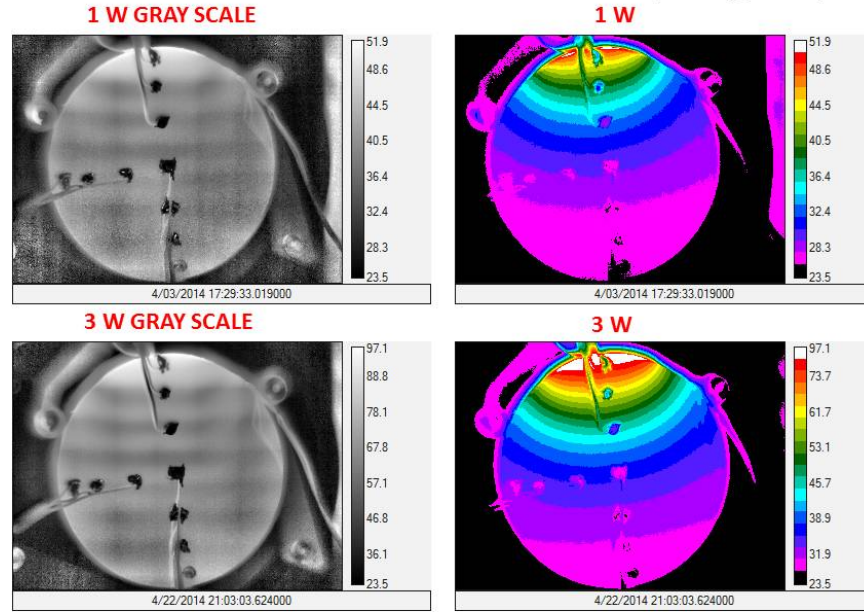


Figure 13. IR images of NBK7 back side.

Table 7. Comparison of Thermocouple and IR Temperatures, NBK7 Back Side.

Temperature (°C)						
TC NO.	TC (1W)	IR ( $\epsilon=0.85$ ) 1W	1W DIFF (TC-IR)	TC (3W)	IR ( $\epsilon=0.85$ ) 3W	3W DIFF (TC-IR)
TC1	44.6	46.3	-1.98	77.17	80.7	-1.3
TC2	37.9	37.7	0.31	58.01	59.4	-0.3
TC3	32.2	32.0	0.64	44.42	43.7	1.0
TC4	26.9	28.2	-0.53	32.94	34.1	-1.6
TC5	27.0	28.0	-0.26	33.01	33.8	-1.2
TC6	26.1	27.4	-0.55	31.17	32.5	-1.8
TC7	25.7	27.0	-0.55	30.26	31.6	-1.9
TC8	25.6	26.3	0.13	29.75	30.1	-0.9
TC9	24.9	25.6	0.17	28.18	28.3	-0.8
TC10	24.4	25.2	0.03	26.99	27.3	-1.1
		Ave.	-0.26		Ave.	-1.0
		STDEV	0.73		STDEV	0.85

The front side images were obtained by rotating the window 180° along the vertical axis as shown in Figure 14, for heater powers of 1W, 2W, and 3W. Again, the window supports do not affect the isotherms. A comparison of thermocouple and IR temperature data is given in Table 7 and shows average temperature differences of  $-0.26 \pm 0.73^\circ\text{C}$  (1W),  $-0.29 \pm 0.6^\circ\text{C}$  (2W), and  $-0.17 \pm 0.9^\circ\text{C}$  (3W).



## BK7 WINDOW FRONT SIDE (deg. C)

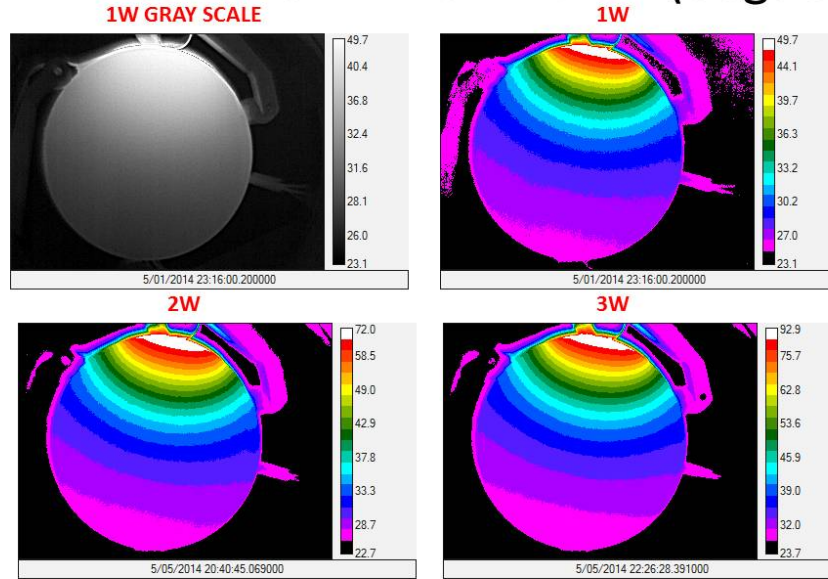


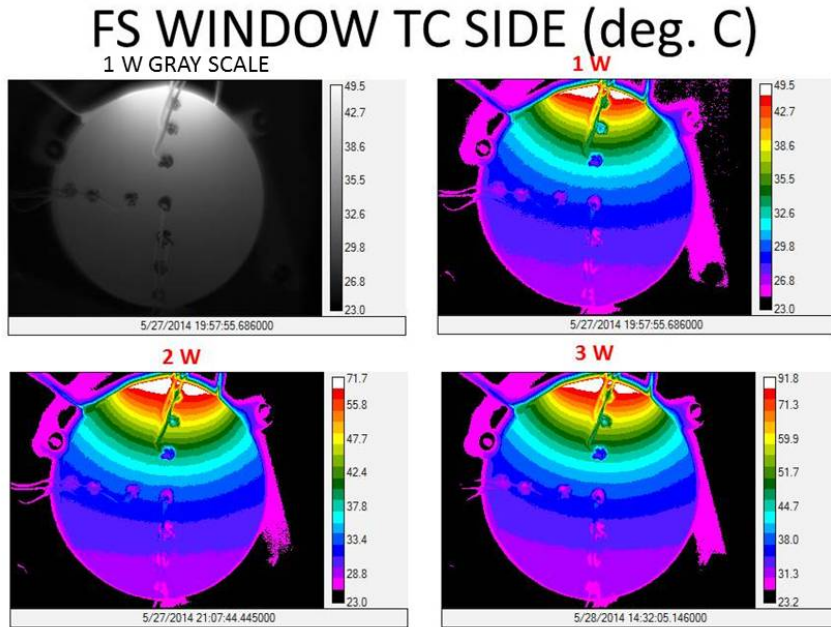
Figure 14. IR images of NBK7 front side.

Table 8. Comparison of Thermocouple and IR Temperatures, NBK7 Front Side.

TC NO.	Temperature (°C)								
	TC (1W)	IR ( $\epsilon=0.85$ ) 1W	1W DIFF (TC-IR)	TC (2W)	IR ( $\epsilon=0.85$ ) 2W	2W DIFF (TC-IR)	TC (3W)	IR ( $\epsilon=0.85$ ) 3W	3W DIFF (TC-IR)
TC1	45.03	45.9	-1.98	64.00	64.7	-0.70	82.59	82.5	0.09
TC2	37.63	38	0.31	49.70	49.2	0.50	61.51	60.5	1.01
TC3	31.96	32.2	0.64	39.05	38	1.05	46.25	44.5	1.75
TC4	26.86	27.9	-0.53	29.54	30.3	-0.76	32.92	33.9	-0.98
TC5	27.12	27.9	-0.26	29.84	30.2	-0.36	33.18	33.7	-0.52
TC6	26.47	27.4	-0.55	28.59	29.3	-0.71	31.48	32.3	-0.82
TC7	26.19	27	-0.55	28.15	28.5	-0.35	30.88	31.2	-0.32
TC8	25.54	26.4	0.13	26.94	27.6	-0.66	29.29	30	-0.71
TC9	24.89	25.5	0.17	25.79	26.1	-0.31	27.66	28	-0.34
TC10	24.34	25.2	0.03	24.83	25.4	-0.57	26.39	27.3	-0.91
		Ave.	-0.26		Ave.	-0.29		Ave.	-0.17
		STDEV	0.73		STDEV	0.60		STDEV	0.90

### 5.3.2 Fused Silica (FS) Optical Material

Both temperature and IR measurements were obtained for the front and back side (thermocouple side) of the FS window for heater powers of 1W, 2W and 3W until steady-state conditions were obtained. IR images are shown in Figure 15 for the back surface. Similar to the NBK7, the infrared emissivity ( $\epsilon$ ) for the camera was set at 0.85, distance from camera to window, 0.33 meters, and air temperature, 20°C. Again, the IR images indicate that the support pads on the window holder did not influence the isotherms, and as expected the highest temperatures are at the top near the heater. The heater power settings provided vertical temperature differences of 27, 48 and 68°C. Table 8 compares the thermocouple and IR temperatures and shows average temperature differences are  $-0.26 \pm 0.62^\circ\text{C}$ ,  $-0.05 \pm 1.05^\circ\text{C}$  and  $0.09 \pm 1.70^\circ\text{C}$  for 1W, 2W and 3W.



**Figure 15. IR images of FS back side.**

**Table 9. Comparison of Thermocouple and IR Temperatures, FS back side.**

TC NO.	Temperature (°C)								
	TC (1W)	IR ( $\epsilon=0.85$ ) 1W	1W DIFF (TC-IR)	TC (2W)	IR ( $\epsilon=0.85$ ) 2W	2W DIFF (TC-IR)	TC (3W)	IR ( $\epsilon=0.85$ ) 3W	3W DIFF (TC-IR)
TC1	43.46	42.9	0.56	60.51	59.3	1.21	76.86	74.2	2.66
TC2	37.53	36.7	0.83	49.05	47	2.05	59.86	56.7	3.16
TC3	32.19	31.9	0.29	38.93	38.2	0.73	45.38	44.1	1.28
TC4	28.17	28.6	-0.43	31.56	31.9	-0.34	34.97	35.3	-0.33
TC5	28.12	28.8	-0.68	31.52	32.4	-0.88	34.59	35.6	-1.01
TC6	27.34	28.3	-0.96	30.4	31.6	-1.2	32.85	34.6	-1.75
TC7	26.82	27.8	-0.98	29.73	30.8	-1.07	31.86	33.3	-1.44
TC8	26.65	27.2	-0.55	28.82	29.3	-0.48	31.02	31.8	-0.78
TC9	25.96	26.2	-0.24	27.67	27.9	-0.23	29.34	29.8	-0.46
TC10	25.38	25.8	-0.42	26.74	27	-0.26	28.03	28.5	-0.47
		Ave.	-0.258		Ave.	-0.047		Ave.	0.086
		STDEV	0.6224		STDEV	1.0543		STDEV	1.6974

The FS front side images are given in Figure 16 and thermocouple and IR comparisons in Table 9. The average temperature differences are  $-0.58 \pm 0.28^\circ\text{C}$ ,  $-0.22 \pm 0.31^\circ\text{C}$  and  $0.09 \pm 0.53^\circ\text{C}$  for 1W, 2W and 3W.

## FS WINDOW FRONT SIDE (deg. C)

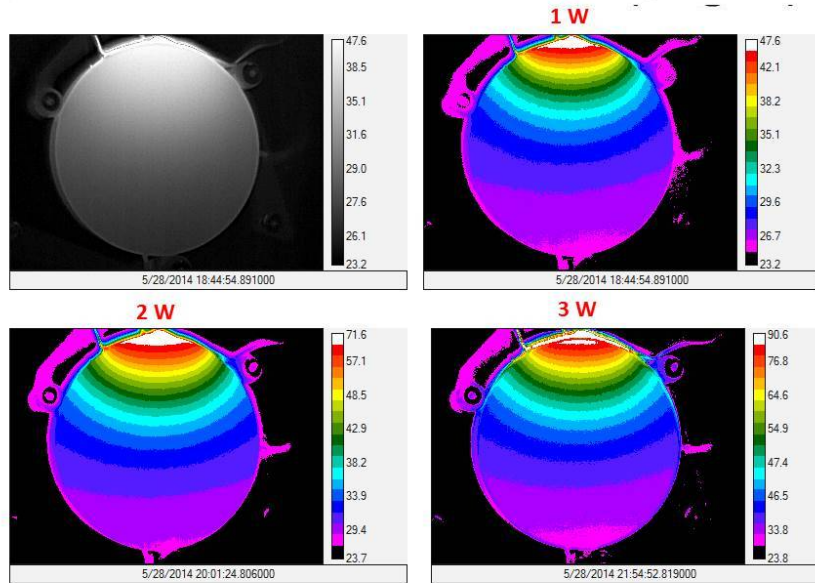


Figure 16. IR images of FS front side.

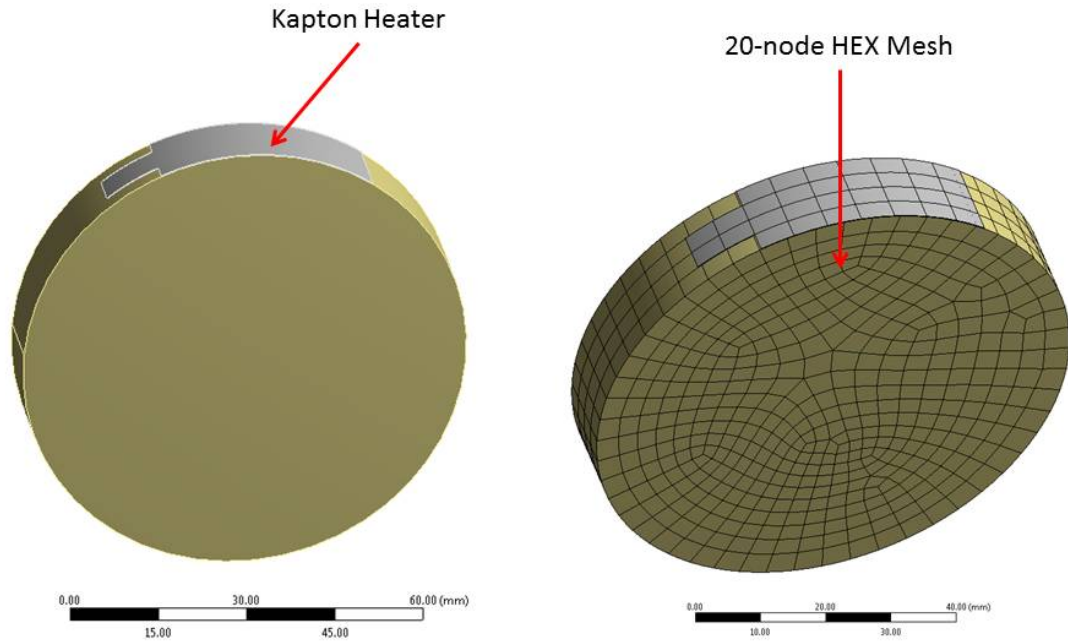
Table 10. Comparison of Thermocouple and IR Temperatures, FS front side.

Temperature (°C)									
TC NO.	TC (1W)	IR ( $\epsilon=0.85$ ) 1W	1W DIFF (TC-IR)	TC (2W)	IR ( $\epsilon=0.85$ ) 2W	2W DIFF (TC-IR)	TC (3W)	IR ( $\epsilon=0.85$ ) 3W	3W DIFF (TC-IR)
TC1	39.73	41.0	-1.27	58.02	57.9	0.12	72.38	71.9	0.48
TC2	35.1	35.6	-0.5	47.8	47.4	0.4	57.85	56.8	1.05
TC3	30.51	31.2	-0.69	38.35	38.6	-0.25	44.29	44.3	-0.01
TC4	27.05	27.7	-0.65	31.36	31.9	-0.54	34.31	34.8	-0.49
TC5	27.06	27.3	-0.24	31.24	31.3	-0.06	34.02	33.9	0.12
TC6	26.35	26.7	-0.35	30.08	30.4	-0.32	32.44	32.6	-0.16
TC7	25.83	26.5	-0.67	29.39	29.6	-0.21	31.38	31.8	-0.42
TC8	25.82	26.4	-0.58	28.84	29.3	-0.46	30.64	31.2	-0.56
TC9	25.29	25.7	-0.41	27.69	28.0	-0.31	29.03	29.2	-0.17
TC10	24.78	25.2	-0.42	26.71	27.3	-0.59	27.69	28.4	-0.71
		Ave.	-0.578		Ave.	-0.222		Ave.	-0.087
		STDEV	0.2849		STDEV	0.306		STDEV	0.533

Comparison of thermocouple and IR measurements show very good agreement for both windows and different heater powers. There are differences between the front and back surfaces after the windows are rotated, due to measurement uncertainties.

## 5.4 Comparison of Thermal Measurements and Model Predictions

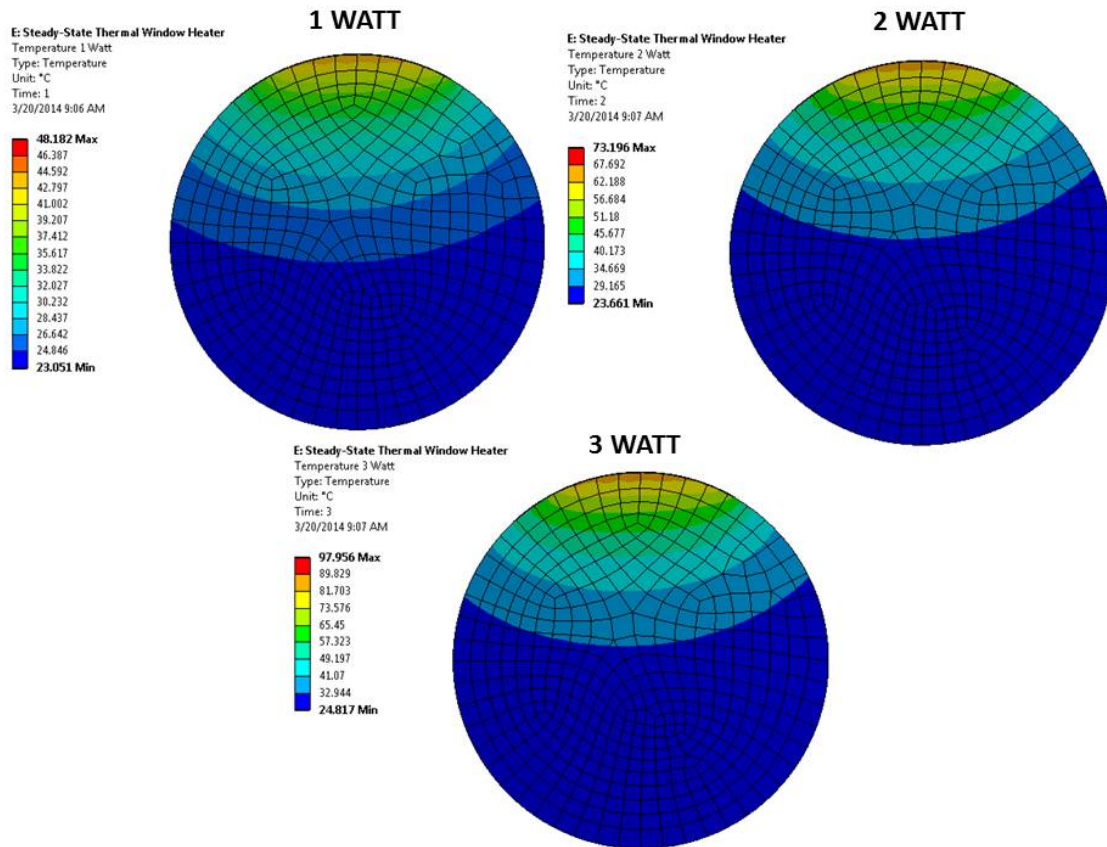
The measured thermocouple temperatures were compared to finite element analysis (FEA) software tools ANSYS/Mechanical and Sierra/Aria. The geometry and FEA mesh for ANSYS is shown in Figure 17 and consisted of a 20-node HEX mesh. The mesh was also used in Sierra/Aria by utilizing the export capability of ANSYS/Fluent and generating an EXODUS-II model geometry. The details are summarized in Appendix A.



**Figure 17. ANSYS window geometry model and 20-node HEX mesh.**

The ANSYS thermal model assumed constant properties, free convection ( $9 \text{ W/m}^2\text{-}^\circ\text{C}$ ), and thermal radiation ( $\epsilon_{\text{window}}=0.8$  and  $\epsilon_{\text{heater}}=0.25$ ). An isothermal plot for the NBK7 window is shown in Figure 18 for 1W, 2W and 3W. Table 11 and Table 12 compare the thermocouple steady-state temperatures and ANSYS model predicts for NBK7 and FS windows. The model results were higher at 1W heater power and lower for 2W and 3W for both windows.





**Figure 18. ANSYS temperature results for NBK7 window.**

**Table 11. ANSYS Model Results and Temperature Data, NBK7 Window.**

	NBK7 Test Results (°C)			Ansys Results (°C)			% Difference (ANSYS-Test)*100/Test		
TC No.	1 Watt	2 Watt	3 Watt	1 Watt	2 Watt	3 Watt	1 Watt	2 Watt	3 Watt
1	42.8	61.0	77.3	43.1	64.8	84.9	0.6	6.3	9.8
2	34.6	45.3	54.7	35.0	49.1	61.7	1.1	8.3	12.8
3	29.4	35.6	40.9	28.6	36.8	43.8	-2.6	3.3	7.1
4	25.6	28.5	30.9	25.4	30.5	34.7	-0.9	7.1	12.3
5	25.7	28.5	30.8	25.1	30.0	33.9	-2.3	5.2	10.2
6	25.1	27.4	29.3	24.6	28.9	32.4	-2.2	5.6	10.5
7	24.8	26.9	28.6	24.2	28.2	31.3	-2.6	4.7	9.3
8	24.4	26.0	27.4	23.7	27.3	30.0	-2.9	4.9	9.5
9	23.9	25.0	25.9	22.8	25.7	27.7	-4.4	2.6	6.8
10	23.4	24.1	24.7	22.5	25.1	26.8	-3.7	4.0	8.5

**Table 12. ANSYS Model Results and Temperature Data, FS Window.**

TC No.	FS Test Results (°C)			Ansys Results (°C)			% Difference (ANSYS-Test)*100/Test		
	1 Watt	2 Watt	3 Watt	1 Watt	2 Watt	3 Watt	1 Watt	2 Watt	3 Watt
1	43.5	60.5	76.9	43.5	63.4	82.2	0.1	4.8	7.0
2	37.5	49.1	59.9	36.4	49.2	61.4	-3.1	0.4	2.5
3	32.2	38.9	45.4	30.7	38.0	44.9	-4.7	-2.5	-1.2
4	28.2	31.6	35.0	27.7	32.1	36.2	-1.8	1.6	3.6
5	28.1	31.5	34.6	27.4	31.6	35.5	-2.5	0.2	2.6
6	27.3	30.4	32.9	26.9	30.6	34.0	-1.6	0.5	3.6
7	26.8	29.7	31.9	26.5	29.8	33.0	-1.1	0.4	3.5
8	26.7	28.8	31.0	26.1	28.9	31.7	-2.2	0.4	2.0
9	26.0	27.7	29.3	25.2	27.3	29.3	-2.8	-1.3	-0.1
10	25.4	26.7	28.0	24.9	26.7	28.4	-1.8	0.0	1.5

Measured data was also compared to Sierra/Aria as described in detail in Appendices B and C. There was good temperature agreement between the two software tools including thermal distortion.

## 5.5 Interferometry Measurements

Interferometry measurements of the optical components were made at each of the prescribed steady-state thermal loads, using a Zygo MST, multi-surface test interferometer. This device incorporates Fourier Transform Phase Shifting interferometry to provide simultaneous measurement of front and back surface figure for plane-parallel optical windows. A measure of optical thickness variation across the aperture is also provided.

Figure 19 shows the baseline measurement of the BK7 window test article at room temperature. The nominal surface figure for both the front and back surfaces, prior to applying thermal load, is less than 60 nm and 70 nm, respectively. The measured optical thickness variation is less than 45 nm.

Figure 20 shows the same component after applying a 3 Watt heat load to the edge of the component for 90 minutes. The surface profiles for the front and back surface are shown to vary by 3.8 microns and 2.7 microns, respectively. Interferometry measurements at the other heat loads are similar in morphology, with smaller variations across the aperture. Note that the optical thickness variation includes effects of the gradient refractive index due to the  $dn/dt$  material properties and the heat induced by the 3 Watt load.

This interferometry data was collected in the last week of the LDRD project, leaving insufficient time to fully reduce the data. With additional funding, the front and back surface interferometry and the temperature measurements will be used to decouple the changes in physical thickness of the part from the heat-induced refractive index gradient, to provide correlation and validation between the measured optical thickness and the model predictions for these parameters.

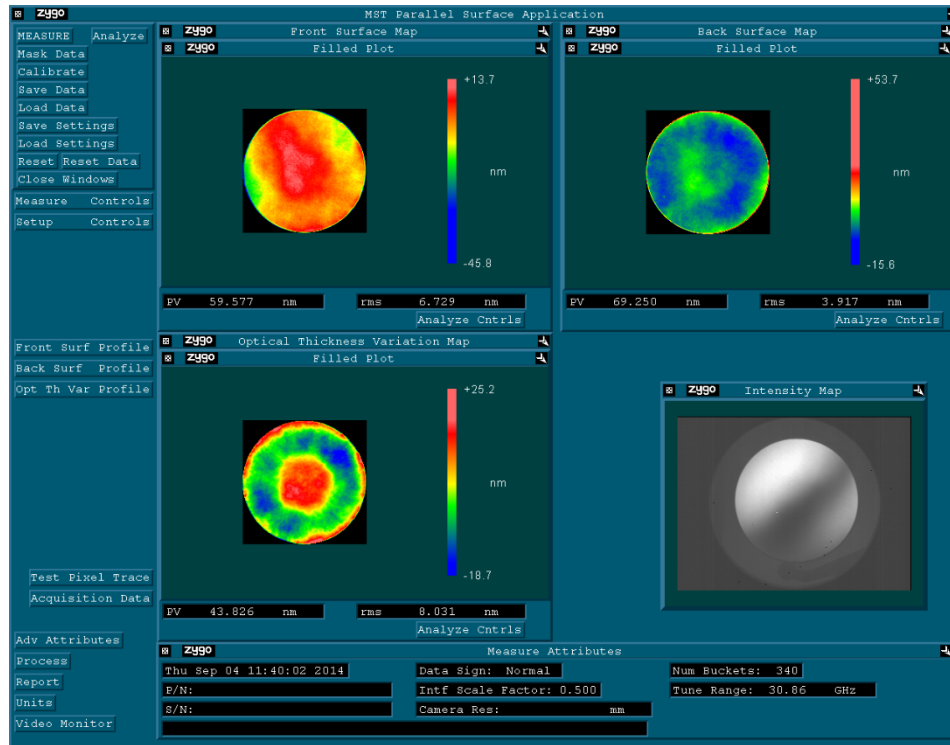


Figure 19. Baseline room temperature interferometry of BK7 test window.

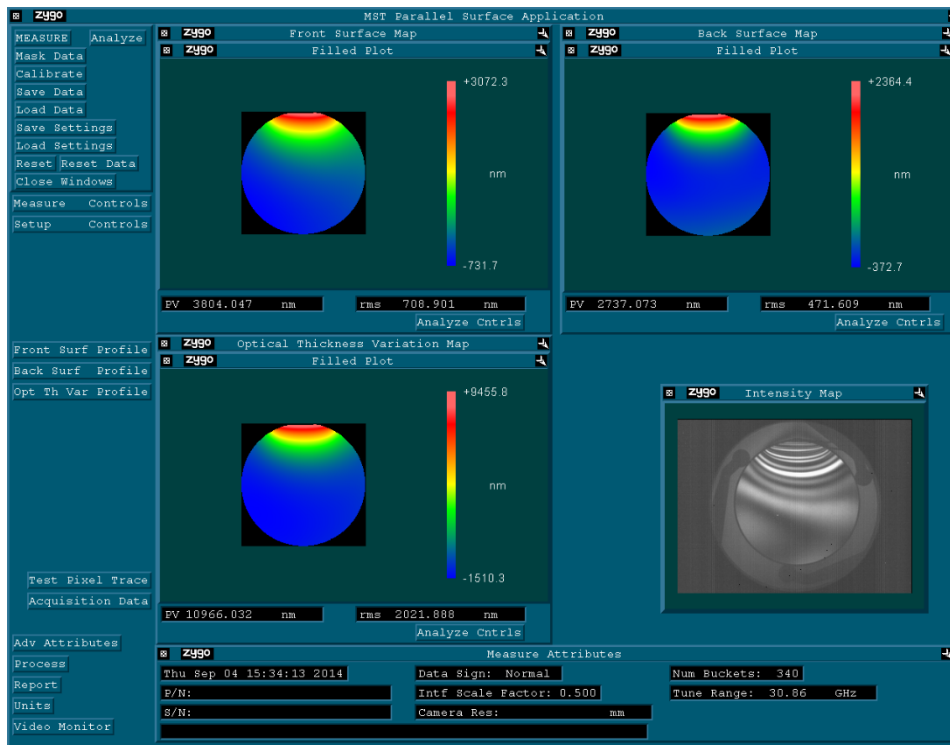


Figure 20. Interferometry of BK7 test window under 3W load for 90 minutes.

## 5.6 SIERRA/Aria Thermal Displacement Predictions for TOP LDRD

A finite element model for lens distortion due to imposed thermal gradients was developed using Aria [13]. Results are presented for the BK7 and Fused Silica lenses with 3 watt heater power. A model discretization based upon an ANSYS mesh was converted to the Genesis format and then used in modeling both lenses. The thermal conditions are approximately those from the 3W tests reported in 5.3. Physical properties for the BK7 and the Fused Silica are summarized in Table 13.

**Table 13: Properties used in analysis.**

Property	FS	BK7	units
Elasticity (E)	7.27E+10	8.20E+10	N·m <sup>-2</sup>
Poisson Ratio ( $\nu$ )	0.160	0.206	-
CTE ( $\alpha$ )	5.70E-06	7.10E-06	K <sup>-1</sup>
Specific heat ( $C_p$ )	741	858	J·kg <sup>-1</sup> ·K <sup>-1</sup>
Thermal conductivity ( $k$ )	1.3	1.114	W·m <sup>-1</sup> ·K <sup>-1</sup>
Density ( $\rho$ )	2201	2510	kg·m <sup>-3</sup>
Zero strain reference temperature	296.34	294.45	K

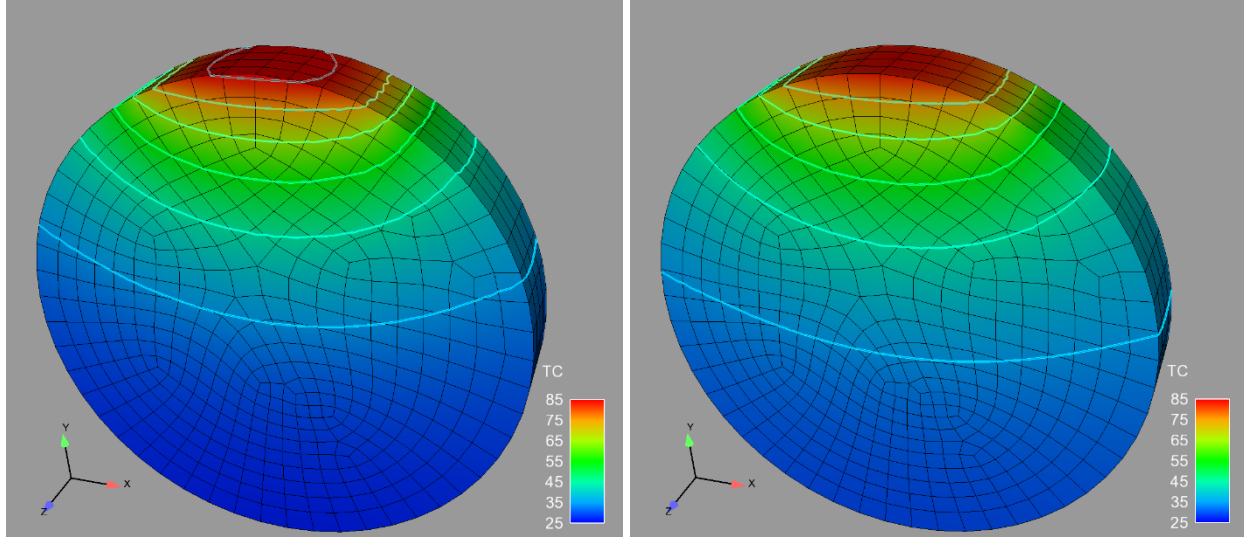
Additional conditions for the analysis are shown in Table 14. Also shown are the zero-strain reference temperatures which were taken to be the reported lab temperatures for the days the experiments were conducted. The detail of the heater is not represented by a meshed volume in the model but its imposed energy flux into the lens is represented by a uniform flux imposed on the set of exterior element faces best approximating the heater location. The 1.75 Watts constitutes 60% of the measured heater power and was observed in other analyses [14] to represent the heating boundary adequately.

**Table 14: Model conditions used in analysis.**

Condition	FS	BK7	units
Constant convective coefficient from lens to ambient air.	5		W·m <sup>-2</sup> ·K
Lens surface emissivity ( $\epsilon$ )	0.5		-
Actual total heat imposed to represent 3W heater	1.75		W
Ambient air temperature	296.34	294.45	K
Zero strain reference temperature	296.34	294.45	K
Element type	HEX20		-
Element Count	1684		-
Node Count	8372		-

Figure 21 shows the predicted steady temperature distributions for the two lenses. Notice that the 85°C contour is absent from the fused silica lens (despite the specified ambient temperature being 1.9°C higher). This result is consistent with the higher (17%) thermal conductivity of fused silica which results in better diffusion of the energy into the lens. Also consistent with the

higher thermal conductivity of the fused silica is the significantly lower position of the 35°C contour.



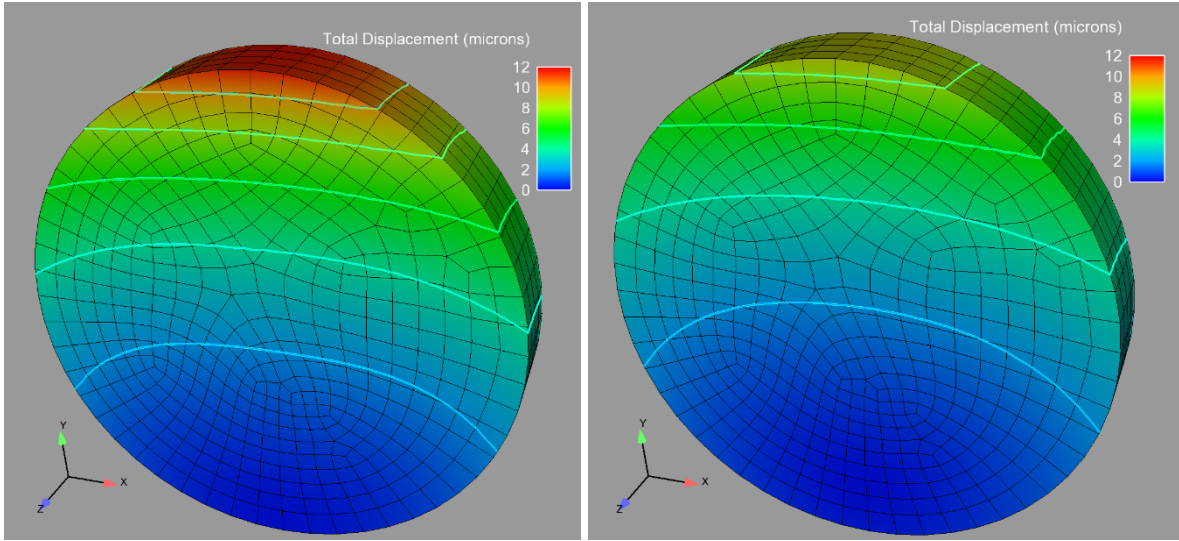
**Figure 21: Predicted steady thermal states for BK7 (left) and Fused Silica (right) lenses. Contours are 35, 45, 55, 65, 75, 85°C.**

The Aria analyses predict the steady temperature distributions and transfer the thermal state of the lens to a structural simulation for calculating the displacement field. This one-way coupling is acceptable because the deformations are small (microns) and do not significantly impact the thermal distribution.

In addition to a reference temperature state corresponding to no strain, the displacement calculation requires boundary conditions to establish/maintain position in the coordinate frame. As oriented in Figure 21, the positive  $x$ -,  $y$ -, and  $z$ -directions are right, up, and out of the page toward the reader. Relative to these directions and a centroid of the unstrained lens at the origin, the boundary conditions were chosen so that:

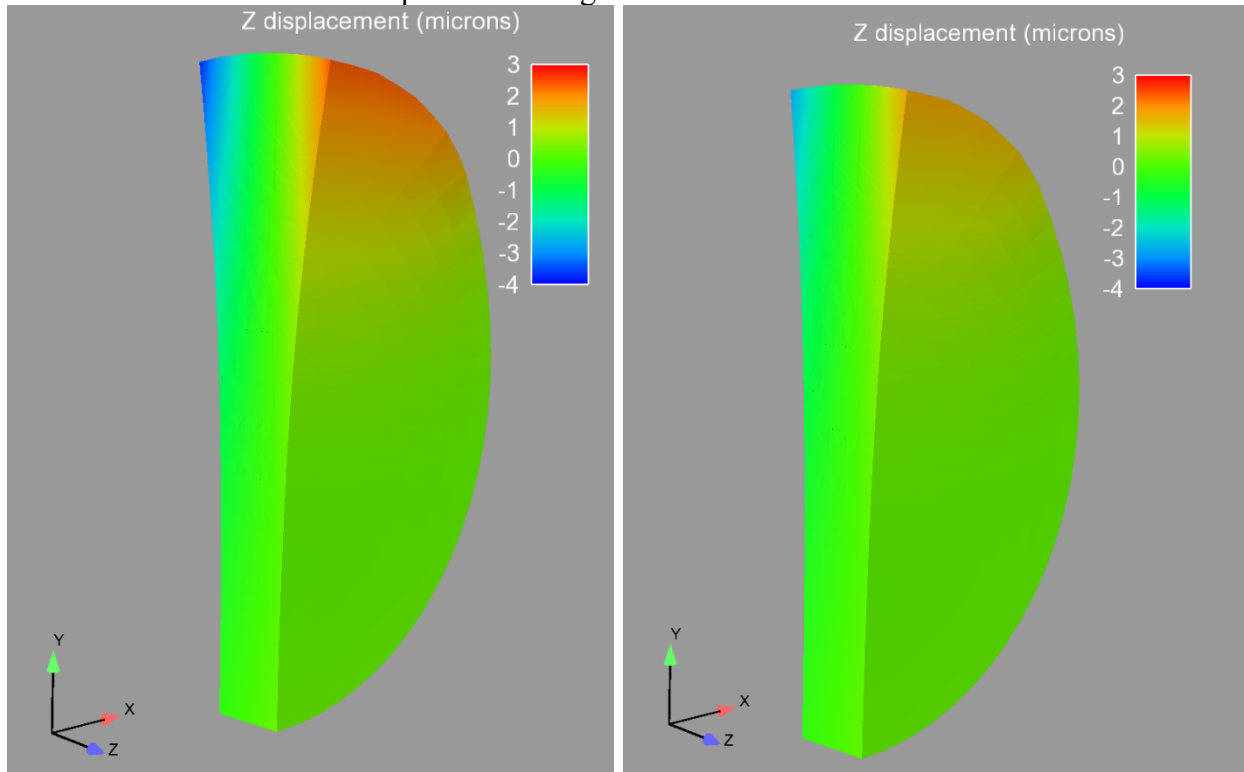
- The points at  $(0, +R, 0)$  and  $(0, -R, 0)$  are stationary in  $x$ .
- The points at  $(+R, 0, 0)$  and  $(-R, 0, 0)$  are stationary in  $y$ .
- The point at  $(0, -R, 0)$  is stationary in  $z$ .

where  $R$  is the undeformed radius of the lens. Figure 22 shows the total predicted displacements for the two materials. The CTE for BK7 being about 25% greater than that of BK7 is evident in the results. The results look to be of the proper magnitude as the thickness of the lens at the hot edge grows approximately the expected  $\alpha H \Delta T$ , where  $H$  is the lens thickness.



**Figure 22: Predicted displacements for BK7 (left) and Fused Silica (right) lenses. Contours are 2, 4, 6, 8, and 10 microns.**

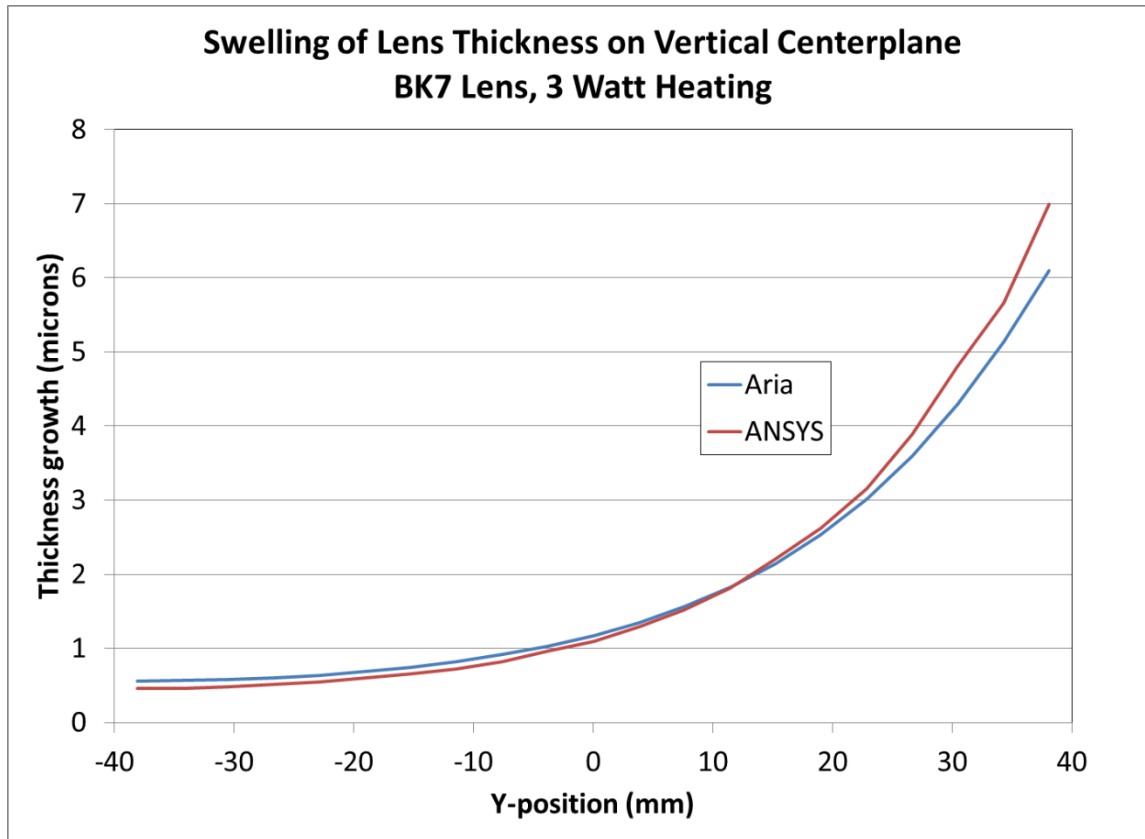
Figure 23 shows a section through the lens colored by the z-displacement while the shape of the lens is shown with the total displacement magnified.



**Figure 23: Z-displacements (microns) as color contours on a lens-shape with exaggerated distortion.**

Figure 24 gives a comparison of the growth of the lens thickness due to 3W heating as indicated by ANSYS [15] [16] and the current predictions. The two thermal analyses targeted agreement

with initial temperature measurements but differ in some of the assumed details and model form. Nonetheless, the predicted z-displacement is in substantial agreement.



**Figure 24: Z-direction “swelling” on x=0 plane comparison of ANSYS [15] [16] and Aria Predictions.**

## 6. CONCLUSIONS

### 6.1 Significant achievements

The theory and results derived in [5] provide strong evidence for the efficacy of the theoretical construct for tracing rays directly in a finite-element volume. The comparison to closed-form gradient-index solutions yields reasonable results within the diffraction limit at optical wavelengths when the maximum gradient profile across individual elements is bounded. The initial method proposed to meet the bounding constraints is to increase mesh density for components with large refractive index gradients, thereby decreasing the local gradient throughout any individual element.

Moreover, the FE ray-trace theory is a new and enabling technology, allowing FE models to be directly imported into a ray-trace code and analyzed over broad spectrum and field-angle parameters with no modification to the base prescription file, and no additional labor imposed by varying the parameters.

To mature the theory into a usable tool requires implementation within a commercial ray-tracing application, validated within the optical design community. This puts the tool into the hands of subject matter experts for broader validation across multiple applications. For this LDRD effort, the Zemax™ ray-trace tool was chosen for its widespread use at SNL, and its open API for implementing user-defined surfaces.

Significant progress was made in developing a Zemax UDS architecture to implement the new ray-trace technique. This code architecture provides hooks into the Zemax menu and pop-up dialog structure to enable additional custom finite-element functions to be inserted into the Zemax suite of capabilities. These new functions include the ability to render a finite-element volume in its native coordinate frame, enabling proper mating of coordinate frame axes into the Zemax master frame. The rendering also provides various color-mapping schemes to indicate gradient temperature profiles, nodal displacements, stress, or strain. And, it provides the capability to visualize mesh geometry, which further enables analyses to correlate localized optical effects with the FEM.

### 6.2 Follow-on effort

#### 6.2.1 *Software Development*

While the UDS software architecture is well founded, the theoretical basis described in 3.4 for mitigating surface interpolation error between the FE methods and the closed-form conic equation were only realized near the end of the LDRD project. As such, schedule did not permit implementation of the construct in the UDS software, and the code is thus incomplete for broad dissemination and use. This code implementation must be accomplished using other funding avenues, either by SNL personnel, or via contracted efforts.



### **6.2.2 Software Validation**

A significant laboratory test and thermal-modeling effort was expended in the course of this project. While the thermal models were tuned and well-validated against the measured temperatures of optical components under prescribed thermal loads, the interferometric measurements of optical performance of these components remain to be compared to the model predictions. Until the UDS software is complete and ready for use, these comparisons cannot be made. Again, this validation comparison must be accomplished using other funding avenues. Even so, performance modeling validation is an on-going process, whereby the modeling tools are applied to many systems of various design in various environments. This endeavor is best achieved through broad dissemination of the tools to subject matter experts in the industry and solicited feedback from the user community.

## **6.3 Future Impact: Coupled Thermo-Optic Transient Analysis**

As stated above, the FE ray-trace theory developed under this LDRD effort is a new and enabling technology. It allows direct optical modeling of various physical environmental loads that directly impact optical performance. Specifically, application of the technology can provide insight to coupled-dynamics problems and issues of material properties prevalent in the fields of high-energy solid-state lasers and short-pulse laser propagation dynamics.

Solid state lasers provide significant improvements in efficiency over other high-energy laser designs, such as chemical or gas lasers. However, the thermal loads induced by the extremely high intra-cavity circulating power are problematic in maintaining a static geometry in the resonant cavity. Thermal expansion of the gain medium and the resonator optics cause the cavity to deform, often causing the resonator to fail after operating only fractions of a second.

The coupled thermo-optical dynamics of the resonant cavity are difficult to model and understand without a fully coupled modeling capability. The cavity design, alignment, and surface figure of the resonator optics serve to establish the initial optical intensity profile within the cavity. Material absorption parameters cause fast-rise thermal transient behavior with corresponding expansion and changes to the index of refraction. These changes impact the performance of the resonant cavity, which then modifies the optical intensity profile and couples with the distortions in the fashion of a feed-back loop.

Other non-linear effects exhibited by propagation of high-energy lasers, such as self-focusing and ablation, are also compounded by the fast-transient, self-induced thermal effects of optical energy through various media. Understanding these coupled dynamics is predicated upon the ability to model the coupled physics of the resonator behavior. The FE ray-trace capability allows the optical performance model to be directly coupled with the thermal-loads model, enabling a fully coupled analysis. Understanding this coupled behavior is a key step in mitigating the thermal effects and realizing high-energy solid state laser designs.

## REFERENCES

1. K. B. Doyle, V. L. Genberg, and G. J. Michels, *Integrated Optomechanical Analysis*. (SPIE Press, 2002), pp. 174-180.
2. B. Richerzhagen, "Finite element ray tracing: a new method for ray tracing in a gradient-index media," *Appl. Opt.* **35**, 6186-6189 (1996).
3. M. Epstein, D. Peter, and M. A. Slawinski, "Combining Ray-Tracing Techniques and Finite-Element Modelling in Deformable Media," *Q. J. Mech. Appl. Math.* **65**, 87-112 (2012).
4. A. Gatej, J. Wasselowski, and P. Loosen, "Thermo-Optical (TOP) Analysis by Coupling FEM and Ray Tracing," in *Proc. SPIE*, 2012, pp. 84290E 1-7.
5. K. N. Schrader, S. R. Subia, J. W. Myre, and K. L. Summers, "Ray tracing in a finite-element domain using nodal basis functions," *Appl. Opt.* **53**, 10-20 (2014).
6. M. Born and E. Wolf, *Principles of Optics*, 6th ed. (Pergamon, 1980), p. 122.
7. A. Sharma, D. V. Kumar, and A. K. Ghatak, "Tracing rays through graded-index media: a new method," *Appl. Opt.* **21**, 984-987 (1982).
8. Zemax Development Corporation, *Zemax User's Guide*. (Zemax Development Corporation, 2007).
9. (2013, April) Schott Optical Glass Datasheets. [Online].  
[http://www.schott.com/advanced\\_optics/us/abbe\\_datasheets/schott\\_datasheet\\_all\\_us.pdf](http://www.schott.com/advanced_optics/us/abbe_datasheets/schott_datasheet_all_us.pdf)
10. (2013) Fused Silica, SiO<sub>2</sub> Glass Properties. [Online]. <http://accuratus.com/fused.html>
11. Minco Polyimide Thermofoil Heaters. [Online].  
<http://www.minco.com/Heaters/~media/WWW/Resource%20Library/Heaters/Polyimide%20Thermofoil%20Heater%20Tech%20Spec.ashx>
12. Omega Ready-Made Insulated Thermocouples. [Online].  
[http://www.omega.com/pptst/5LSC\\_5SRTC.html](http://www.omega.com/pptst/5LSC_5SRTC.html)
13. "SIERRA Multimechanics Module," in *Aria User Manual*.
14. M. Larsen, "UQ Analysis of Lens Thermal Modeling," (1514), Sandia National Laboratories, Albuquerque, NM, Internal Memo 2014.
15. E. Couphos, 140606\_BK7\_window\_heater\_3Watt\_coarse\_mesh.txt, 2014, Exported ANSYS FE Data file of laboratory experiment.
16. E. Couphos, 140606\_BK7\_window\_heater\_3Watt\_fine\_mesh, 2014, Exported ANSYS FE Data file of laboratory experiment.



## APPENDIX A: THEORETICAL BASIS

# Ray tracing in a finite-element domain using nodal basis functions

Karl N. Schrader,\* Samuel R. Subia, John W. Myre, and Kenneth L. Summers

Sandia National Laboratories, P.O. Box 5800, Albuquerque, New Mexico 87185, USA

\*Corresponding author: kschrader@sandia.gov

Received 13 March 2014; revised 14 April 2014; accepted 17 July 2014;  
posted 18 July 2014 (Doc. ID 208169); published 15 August 2014

A method is presented for tracing rays through a medium discretized as finite-element volumes. The ray-trajectory equations are cast into the local element coordinate frame, and the full finite-element interpolation is used to determine instantaneous index gradient for the ray-path integral equation. The finite-element methodology is also used to interpolate local surface deformations and the surface normal vector for computing the refraction angle when launching rays into the volume, and again when rays exit the medium. The procedure is applied to a finite-element model of an optic with a severe refractive-index gradient, and the results are compared to the closed-form gradient ray-path integral approach. © 2014 Optical Society of America

OCIS codes: (080.5692) Ray trajectories in inhomogeneous media; (120.6810) Thermal effects.  
<http://dx.doi.org/10.1364/AO.53.000F10>

### 1. Introduction

The subject of ray tracing through inhomogeneous refractive media has been addressed at varying levels of detail for a number of decades. Early publications [1–4] concentrated on generalized extension of ray-trace theory within a known gradient field represented by a closed-form equation. This theory is well formed in terms of differential calculus, and with the change of variable introduced by Sharma *et al.* [3], it can be easily cast into a state-space notation. More recent publications have concentrated on application of the theory using numerical methods in a discretized, or finite-element, volume, which lends itself well to computer analysis of optical propagation through pseudo-randomly distributed gradient volumes [5–8]. These latter approaches begin to integrate and exploit the rich and mature finite-element numerical methods developed by analysts in the fields of structural mechanics and heat transfer. Indeed, Gatej *et al.* [8] developed an approach to couple commercial finite-element

method (FEM) and ray-tracing simulation software via an intermediate processing step.

This paper extends these approaches by fully integrating the finite-element numerical methodology to enable the tracing of rays directly within a discretized volume. The approach is most similar to that presented by Richerzhagen [6], with the exception that the refractive index is not assumed to be constant (homogeneous) throughout the discrete element, but rather varies continuously across the element volume with exact values defined at nodal positions and a formalized set of interpolation “shape functions” to derive values at intermediate points within the volume. In this approach, the generalized gradient-index ray-trace equations are first transformed in terms of the local coordinate frame of a solid (3D) element. The shape function of the element is then used to calculate the local index gradient, and the resulting system of equations is cast into state-space form for implementation by a numerical solver.

The approach outlined in this paper may be implemented in a commercial ray-trace application through a customization API currently provided by most software vendors [9,10]. This eliminates the need for intermediate processing and analysis to

---

1559-128X/14/240F10-11\$15.00/0  
© 2014 Optical Society of America

enable ray tracing through the inhomogeneous medium. Software can be written to directly read a finite-element grid file and trace rays through the volume upon request by the host application. This can potentially be used to fully couple thermal and ray-trace analyses, enabling detailed study of intra-cavity thermodynamic effects in high-energy lasers, or the performance impact of thermal loads on space-borne optical systems. The approach is also extensible to analyzing optical performance in the presence of any number of other distributed material properties.

## 2. State-Space Ray-Trace Equations in a Gradient-Index Medium

### A. Mathematical Basis

The differential ray equation is given in [11] as

$$\frac{d}{ds} \left[ n(\mathbf{r}) \frac{d\mathbf{r}}{ds} \right] = \nabla n(\mathbf{r}), \quad (1)$$

where  $\mathbf{r}$  is the position vector of a point on the ray,  $n(\mathbf{r})$  is the refractive index of the medium at position  $\mathbf{r}$ , and  $ds$  is a differential portion of the arc length  $s$  along the ray.

Sharma *et al.* [3] introduced a change of variable in this equation, to form

$$\frac{d^2 \mathbf{r}}{dt^2} = n \nabla n = \frac{1}{2} \nabla n^2 = \frac{1}{2} \begin{bmatrix} \frac{\partial n^2}{\partial x} \\ \frac{\partial n^2}{\partial y} \\ \frac{\partial n^2}{\partial z} \end{bmatrix}, \quad (2)$$

where  $t \equiv \int (ds/n)$  and  $dt \equiv (1/n)ds$ . In Eq. (2), the explicit dependence of refractive index on position has been dropped for convenience, this dependency being implied by the gradient operator. To be complete, it is also noted that Sharma *et al.* defined an optical ray vector  $\mathbb{T}$  as

$$\mathbb{T} \equiv \frac{d\mathbf{r}}{dt} = n \frac{d\mathbf{r}}{ds}. \quad (3)$$

The implications of these changes have been described in detail [3] and will not be further examined here. Instead, the new formulation will be used to enable an abstracted concept of the ray trajectory based upon a Newtonian dynamics approach.

While the change of variable  $t$  introduced by Sharma *et al.* is literally a dummy variable of integration, one can easily abstract upon the concept of  $t$  as an independent “pseudo-time” variable. In this context, an abstract perspective of the optical ray vector  $\mathbb{T}$ , defined in Eq. (3), can be obtained, namely that  $\mathbb{T}$  is the “pseudo-velocity” of a point  $\mathbf{r}$  on the ray, i.e., the first derivative of position  $\mathbf{r}$  w.r.t.  $t$ . Likewise, Eq. (2) can be abstracted to represent the “pseudo-acceleration” of a point  $\mathbf{r}$  on the ray. This abstraction allows assembly of these “equations of motion” into a state-space system of equations, from which

an elementary coordinate transformation can be derived to enable direct application within the finite-element domain.

To elaborate, we expand Eq. (2) into 3D space to yield

$$\frac{d^2 \mathbf{r}}{dt^2} = \frac{d^2 x}{dt^2} \hat{i} + \frac{d^2 y}{dt^2} \hat{j} + \frac{d^2 z}{dt^2} \hat{k} = \frac{1}{2} \nabla n^2. \quad (4)$$

At this point, it is useful to introduce a shorthand notation  $\{\cdot\}$ , which denotes the derivative of a state variable w.r.t.  $t$ , i.e.,

$$\{\cdot\} = \frac{d}{dt}.$$

We then define a state vector  $\bar{\mathbf{x}}$ , composed of the position and velocity of point  $\mathbf{r}$ , in 3D space:

$$\bar{\mathbf{x}} \equiv \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \quad (5)$$

Finally, we may combine Eqs. (2), (4), and (5) into a full state-space representation of the differential equations of motion for a point that lies on the ray,

$$\dot{\bar{\mathbf{x}}} = \begin{bmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 0_3 \\ I_3 \end{bmatrix} \frac{1}{2} \nabla n^2, \quad (6)$$

where  $0_3$  denotes a square zero matrix of dimension 3 and  $I_3$  is the  $3 \times 3$  identity matrix. In this equation, the gradient of the index acts as a “forcing function” that imparts a “pseudo-acceleration” on the ray as it travels through the medium.

Integration of this set of differential Eq. (6) w.r.t. the independent “pseudo-time” variable yields the ray trajectory in a standard Cartesian coordinate frame. Such integration requires knowledge of the gradient-index variation as a function of position, either by a closed-form equation, or via an interpolation algorithm applied to points within the volume where the refractive index is known. The finite-element methodology provides a useful framework for the latter approach.

### B. Finite-Element Methodology

Finite-element (FE) theory is used in many engineering fields to reduce complex problems of partial differential equations to a numerically tractable form. Experts in the fields of computational mechanics have developed a rich theoretical basis for discretizing volumes and surfaces into collections of nodes that are grouped into discrete finite elements.

The FE methodology also provides functions for interpolation of geometry and for other parameters (e.g., displacement, temperature) defined at the nodes, as well as for calculating gradients of these



parameters within the space defined by the element. These functions are understandably dependent upon the geometry of the particular element, i.e., how many nodes are contained and the “shape” of the element (triangular, quadrilateral, tetrahedron, hexahedron, etc.) to which they belong. However, at the most basic level, all these functions are derived via a mapping of the node geometry in physical Cartesian space to a unit-normalized geometry of coordinates referenced to a local origin within each element. For example, a 20-node hexahedron element extracted from a discretized FE mesh of an optical component may have the physical shape shown in Fig. 1(a). However, in the local coordinate frame  $(\xi, \eta, \zeta)$  of Fig. 1(b), it is mapped to a perfect cube, centered on the origin, and bounded in all dimensions within the range  $[-1, 1]$ .

Before continuing, it is important to note some key attributes of this mapping. First, there exists a transformation between the local coordinates and the physical coordinates (although the inverse transform may be ill posed; see Appendix B). For the element shown in Fig. 1, this transform contains second-order powers of  $(\xi, \eta, \zeta)$  (imagine fitting a quadratic equation through any three points that make up an edge of the element). Second, every element contained in the optical component shown on the left side of Fig. 1(a) has its own coordinate frame. Therefore,

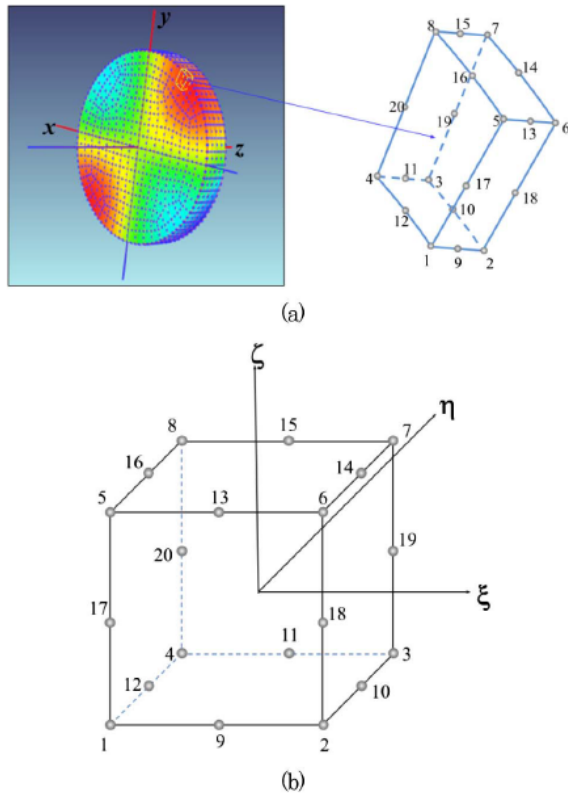


Fig. 1. HEX20 element in (a) physical coordinate space and in (b) local coordinate space.

the mapping of local coordinates to physical coordinates (and *vice versa*) is unique for each element within the volume. Third, the nodes of an element are numbered in a specific order (see Appendix A) that defines the orientation of the local coordinate axes within the element, and upon which the interpolation and other methods of the FE construct rely. This ordering must be maintained for any analyses that utilize these methods. And finally, neighboring elements share the nodes of their neighboring faces.

While a complete treatise on finite-element theory is beyond the scope of this paper, these important aspects should be retained for this discussion. The reader is referred to the literature [12,13] for further description of finite-element theory.

The remainder of this paper provides a theoretical basis for casting the physical ray-trace Eq. (6) into the local coordinate frame  $(\xi, \eta, \zeta)$  of a finite element. This new formulation allows direct integration within the  $(\xi, \eta, \zeta)$  domain using FE methods to yield the ray trajectory within the curvilinear space of the local coordinate frame. Within this frame, element face crossings are easily found, due to the bounding  $[-1, 1]$  geometry defined by the element. At the interfaces of neighboring elements, a simple coordinate transformation may be applied to cast the ray into the new local coordinates of the neighboring element. In this way, a ray may be traced through the optical component, element by element, until it exits the discretized volume.

### C. Ray-Trace Equations in Local Element Coordinates

To cast the ray-trace equations into the FE local element coordinate frame requires each reference to  $(x, y, z)$  in Eq. (6) (whether explicit, or implied) be transformed to  $(\xi, \eta, \zeta)$  coordinates. Beginning with the gradient of the squared refractive index, we expand the notation to explicitly show dependence on position  $(x, y, z)$ , namely

$$\nabla n^2 = \begin{bmatrix} \frac{\partial n^2}{\partial x} \\ \frac{\partial n^2}{\partial y} \\ \frac{\partial n^2}{\partial z} \end{bmatrix}, \quad (7)$$

where  $n$  is a function of position, i.e.,  $n = n(x, y, z)$ .

Transformation of the gradient into  $(\xi, \eta, \zeta)$  space is accomplished via the Jacobian matrix of the element,  $\mathcal{J}$ , as follows:

$$\mathcal{J} \begin{bmatrix} \frac{\partial n^2}{\partial x} \\ \frac{\partial n^2}{\partial y} \\ \frac{\partial n^2}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial n^2}{\partial x} \\ \frac{\partial n^2}{\partial y} \\ \frac{\partial n^2}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial n^2}{\partial \xi} \\ \frac{\partial n^2}{\partial \eta} \\ \frac{\partial n^2}{\partial \zeta} \end{bmatrix}, \quad (8)$$

where formulation of the Jacobian matrix is defined by the FE methodology [13] (see Appendix A). Note that  $\mathcal{J}$  is a function of position in the local coordinate frame,  $\mathcal{J} = \mathcal{J}(\xi, \eta, \zeta)$ , and the position dependency of  $n$  is also mapped to  $(\xi, \eta, \zeta)$  by the FE shape functions.

A similar approach is used to transform the “pseudo-velocity” vector into local coordinates, as shown in Eq. (9):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{d\xi}{dt} \\ \frac{d\eta}{dt} \\ \frac{d\zeta}{dt} \end{bmatrix} = \mathcal{J}^T \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix}. \quad (9)$$

The “pseudo-acceleration” vector is transformed by first noting that it is the derivative of “pseudo-velocity” w.r.t.  $t$ . This derivative operation is applied to Eq. (9) to yield

$$\begin{aligned} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \frac{d}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \frac{d}{dt} \left\{ \mathcal{J}^T \begin{bmatrix} \frac{d\xi}{dt} \\ \frac{d\eta}{dt} \\ \frac{d\zeta}{dt} \end{bmatrix} \right\} \\ &= \frac{d\mathcal{J}^T}{dt} \begin{bmatrix} \frac{d\xi}{dt} \\ \frac{d\eta}{dt} \\ \frac{d\zeta}{dt} \end{bmatrix} + \mathcal{J}^T \begin{bmatrix} \frac{d^2\xi}{dt^2} \\ \frac{d^2\eta}{dt^2} \\ \frac{d^2\zeta}{dt^2} \end{bmatrix} = \frac{d\mathcal{J}^T}{dt} \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix} + \mathcal{J}^T \begin{bmatrix} \ddot{\xi} \\ \ddot{\eta} \\ \ddot{\zeta} \end{bmatrix}. \end{aligned} \quad (10)$$

All of the components are now available to cast the ray equations into the  $(\xi, \eta, \zeta)$  domain. By substituting, rearranging, and assembling from Eqs. (6)–(10), we derive the ray “acceleration” expression

$$\begin{bmatrix} \ddot{\xi} \\ \ddot{\eta} \\ \ddot{\zeta} \end{bmatrix} = \mathcal{J}^{-T} \left\{ \frac{1}{2} \mathcal{J}^{-1} \begin{bmatrix} \frac{\partial n^2}{\partial \xi} \\ \frac{\partial n^2}{\partial \eta} \\ \frac{\partial n^2}{\partial \zeta} \end{bmatrix} - \frac{d\mathcal{J}^T}{dt} \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix} \right\}, \quad (11)$$

cast purely in terms of the  $(\xi, \eta, \zeta)$  domain. Full expansion of each component using the FE construction of the Jacobian, its derivative, and the gradient is given in Appendix A.

It is worth pausing at this juncture to note some important aspects of Eq. (11). It is the analog of Eq. (2), cast in the  $(\xi, \eta, \zeta)$  domain. However, there are two forcing function components on the right side of Eq. (11), whereas only one exists in Eq. (2). The left-most bracketed term of Eq. (11) is seen to be proportional to the gradient of  $n^2$  in the  $(\xi, \eta, \zeta)$  domain, in keeping with Eq. (2) in the physical domain. However, the second bracketed component carries a dependency on both velocity and the rate of change of the Jacobian, which has no apparent analog in the physical domain.

Introduction of this component is easily understood by considering the mapping transformation from physical to local coordinates. For the special case of a perfectly homogeneous medium, the gradient of  $n^2$  in the physical domain is zero, and the resulting ray trajectory is linear. However, when mapping the “skew aspect” of the cube in physical coordinates [Fig. 1(a)] to the cubic volume [Fig. 1(b)] in local coordinates, the space within the element must necessarily become a curvilinear space. In

short, a straight ray path in physical coordinates is generally a curved ray path in local coordinates. This curvature implies a “lateral acceleration” of the ray in local coordinates, caused purely by the coordinate transformation. In this context, the significance of the second bracketed term of Eq. (11) becomes clear.

The magnitude of the second term is wholly dependent on the skew aspect of the element within the physical coordinate space. If the element shape is a perfect cube in physical coordinates, the mapping to local coordinates is linear and the Jacobian of the element will be constant (zero derivative), thus removing any dependency on “velocity” within the local coordinate frame.

#### D. Inter-Element Ray Transitions within the Volume

At the interfaces between elements, the ray equations must transition from the local coordinate frame of the terminal element to that of the entrance element. While one might consider transforming to physical coordinates and ray direction cosines as an intermediate step in this process, it is beneficial to exploit aspects of the geometry to obtain a single-step transform, which increases computational efficiency and reduces the potential for both implementation errors and accumulated numerical errors.

As stated previously, neighboring elements share the nodes of their neighboring faces. In the FE domain, this abutment of neighboring elements can be pictured as two cubes (for hexahedron elements), which share a face, but which are centered on separate element coordinate frames. This is graphically depicted in Fig. 2, where an arbitrary orientation of the two local coordinate frames has been chosen for illustrative purposes.

In general, the local coordinate frames of each element are independently oriented according to the node-ordering scheme assigned by the FE grid generation software. Note that shared nodes on the neighboring faces are assigned different node numbers within each element. Indeed, this is

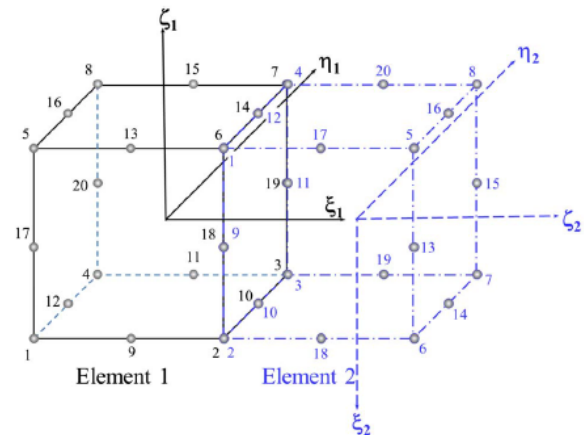


Fig. 2. Neighboring hexahedron elements in the finite-element domain.





required for a consistently ordered numbering scheme within coordinate frames that maintain a right-handed orientation. For the case shown, it is seen that element 1 shares its nodes numbered [2, 3, 7, 6, 10, 19, 14, 18] on the  $\xi_1 = +1$  face, with element 2 nodes numbered [2, 3, 4, 1, 10, 11, 12, 9] on the  $\zeta_2 = -1$  face. These are physically the same nodes, with the same properties (temperature, refractive index, etc.), and are simply assigned a different number within each element to maintain the consistency of the FE construct.

Consider for a moment a ray traveling through element 1 and incident upon its  $\xi = +1$  face, such that it exits the element and enters element 2, on its  $\zeta = -1$  face. As the ray transitions from element 1 to element 2, it crosses the element boundary, and its orientation must be transformed into the new frame. This transformation involves both the local master element coordinates and the ray direction at the interface. Note that, although the refractive medium is continuous at the boundary and no physical refraction or reflection occurs, the curvilinear mappings of the two frames are completely independent and may cause an apparent discontinuity in ray direction across the boundary in the abutted local coordinate spaces. However, when converted to physical coordinates, any such discontinuity is rectified by the individual FE transformations of the two elements.

In the plane of transition shared by the neighboring elements, the ray coordinates maintain the same absolute values in both frames, and are merely transformed by the relative rotation of the axes, i.e.,

$$\begin{bmatrix} \xi_2 \\ \eta_2 \\ \zeta_2 \end{bmatrix} = T \begin{bmatrix} \xi_1 \\ \eta_1 \\ \zeta_1 \end{bmatrix}, \quad (12)$$

where  $T$  is the transformation matrix. An algorithm to derive a generalized transformation matrix between any two arbitrarily oriented neighboring elements is given in Appendix C.

The change in the local coordinate ray direction as it transitions from element 1 to element 2 is derived from Eq. (9). Since the “pseudo-velocity” (i.e., ray direction) in physical coordinates is continuous at the boundary, we deduce that

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathcal{J}_1^T \begin{bmatrix} \dot{\xi}_1 \\ \dot{\eta}_1 \\ \dot{\zeta}_1 \end{bmatrix} = \mathcal{J}_2^T \begin{bmatrix} \dot{\xi}_2 \\ \dot{\eta}_2 \\ \dot{\zeta}_2 \end{bmatrix}, \quad (13)$$

which can be rearranged to yield

$$\begin{bmatrix} \dot{\xi}_2 \\ \dot{\eta}_2 \\ \dot{\zeta}_2 \end{bmatrix} = \mathcal{J}_2^{-T} \mathcal{J}_1^T \begin{bmatrix} \dot{\xi}_1 \\ \dot{\eta}_1 \\ \dot{\zeta}_1 \end{bmatrix}. \quad (14)$$

Note the Jacobian matrices,  $\mathcal{J}_1$  and  $\mathcal{J}_2$ , are derived at the point of transition between the separate local

coordinate frames of element 1 and element 2, respectively. Also note that this formulation bypasses direct calculation of the ray direction cosines in physical space, which would involve interpolating the refractive index at the boundary and possibly introduce additional errors that could accumulate at successive element interfaces as the ray propagates through the FE volume.

### 3. Ray-Trace Algorithm

A basic set of assumptions must be fulfilled prior to launching the ray-trace algorithm. These include the following:

A. The incident (incoming) ray definition is complete and available, including the following:

- i. The intersection of the ray with a known reference plane in physical coordinates.
- ii. The ray direction cosines w.r.t. the physical coordinate axes.
- iii. The wavelength of the representative ray (if the FE medium is dispersive).

B. The refractive index of the ambient medium at the point of incidence is known.

C. The full set of nodes for the optical component and their associated properties are available, including the following:

- i. Node ID number.
- ii. Nominal position in physical coordinates.
- iii. Displacements due to load conditions, in physical coordinates.
- iv. Refractive index squared at the node. This can be a constant, or a function of material properties, wavelength, and/or any other nodal parameters, including position and temperature.
- v. Nodal values for any parameters required to compute the refractive index at the node (temperature, stress, etc.).
- vi. A list of element ID numbers for each element to which the node belongs.

D. The full set of elements making up the optical component are available, including the following:

- i. Element ID number.
- ii. The element type (hexahedral, tetrahedral, etc.).
- iii. The number of nodes contained in the element.
- iv. The node ID numbers of each member node, correlated to the ordered node number within the element.

E. List of nodes (node ID numbers) that lie on the front surface.

F. List of nodes that lie on the back surface.

Once these data have been registered, the FE ray-trace algorithm may proceed. The general approach is different from that outlined in [5], since the Jacobian of solid elements required to model the transmissive optical component is a function of position. The algorithm proceeds as follows:

1. Locate the element the ray strikes. A multitude of conceivable methods to achieve this goal certainly exist. Choice of method should include the user's requirement for computational efficiency and memory storage. The method implemented for this paper is as follows:

(a) Estimate the ray incidence point by calculating the intersection of the ray with the nominal front surface in physical coordinates. If the surface is a conic of revolution, this can be accomplished using the classic technique of solving for ray intersection given a ray reference point, the direction cosines, and the conic equation.

(b) From the list of front-surface nodes, query each node position to find the nearest nodes to the estimated ray intersection point. A brute-force search is sufficient, although some advanced algorithms may prove more efficient.

(c) For each nearest node, query the list of elements to which it belongs, and find the set intersection of this list with those of the other nearest nodes. This set intersection quickly eliminates elements that do not contain all the nearest nodes. For the HEX20 elements used in the example FE model, it was found that the critical number of nearest nodes is two. This guarantees at least one candidate element will be found, and possibly more. For certain skew geometries, it was found that three nearest nodes might not belong to one common element, resulting in an empty set intersection.

(d) Determine whether the ray strikes the first element returned in step (c). Again, a number of methods to make this determination may exist. The method used for this paper is a modification of the point-in-polygon algorithm commonly used in computational solid geometry (CSG) applications. Another method would be to convert the physical coordinates to local coordinates in the element of interest, as shown in Appendix B, and determine whether these local coordinates are bounded in the range  $[-1, +1]$ .

(e) If it is determined that the ray does not strike this element, compute an error direction and iterate from step (d) for the neighboring elements, until the incident element is found.

2. Convert the physical coordinates of the intersection point to local coordinates in the incident element, as shown in Appendix B. Note that the local coordinates of the intersection point must be bounded in the range  $[-1, +1]$ , in each axis.

3. Interpolate the refractive index of the element at the intersection point, using the element shape functions (see Appendix A).

4. Calculate the surface normal at the point of intersection.

5. Determine the ray refraction (in physical coordinates) at the point of incidence, using the ray direction cosines, the ambient refractive index, the element refractive index, and the surface normal at the point of intersection.

6. Calculate the Jacobian of the element at the intersection point (see Appendix A).

7. Determine the ray "pseudo-velocity" (in local coordinates) using Eqs. (3) and (9), to form

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix} = \mathcal{J}^{-T} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathcal{J}^{-T} n \begin{bmatrix} \cos \alpha \\ \cos \beta \\ \cos \gamma \end{bmatrix}, \quad (15)$$

where the vector terms are the direction cosines of the refracted ray (in physical coordinates) at the point of incidence.

8. Integrate the ray trajectory through the FE, using Eq. (11), until it intersects an element face, defined by a  $\xi$ ,  $\eta$ , or  $\zeta$  value of  $[-1, +1]$ .

9. Determine whether the terminal face of the element has a neighboring element, and

(a) if it does have a neighboring element (ray is entering another element), derive the element-transition ray transformation as outlined in the previous section, and iterate from step 8.

(b) if it does not have a neighboring element (ray is exiting the FE volume),

i. use the element shape functions to interpolate the refractive index and the exit position in physical coordinates.

ii. calculate the ray direction cosines of the exiting ray (inside the element, and in physical coordinates) by inverting Eq. (15), to yield

$$\begin{bmatrix} \cos \alpha \\ \cos \beta \\ \cos \gamma \end{bmatrix} = \frac{1}{n} \mathcal{J}^T \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix}. \quad (16)$$

iii. Determine the refraction of the exiting ray (in physical coordinates), using the same method of step 5.

This algorithm is generalized for a nonsequential ray-trace capability. As written, under certain circumstances rays may enter and exit the same surface, or exit through edges of a component. For application within a sequential ray-trace engine, these cases must be monitored to cull rays that do not encounter each surface in a sequential fashion.

#### 4. Implementation Results

The algorithm described in Section 3 was implemented using MATLAB tools and scripting language. The discretized FEM of a 152.4 mm diameter by 19.05 mm thick optical window, shown in Fig. 1, was created using the ANSYS Workbench finite-element modeling application. A script was created to read an ANSYS ASCII output file directly into the MATLAB workspace. The FEM ray trace was then performed on the discretized volume.

To verify performance of the algorithm, a comparison method similar to that described in [6] was implemented. In this approach, a closed-form equation for a gradient refractive index is used to generate index values at each node. The FEM ray-trace





results are then compared directly to the closed-form ray-trace solution.

For the purpose of this comparison, the closed-form expression used for the refractive index was chosen to be

$$n^2 = 2.5 - 1.55 \times 10^{-4}(x^2 + y^2), \quad (17)$$

where  $x$  and  $y$  are in units of millimeters. This creates an arbitrary, but severe, radial gradient profile ranging from 1.265 at the edge of the window to 1.581 at the center. Note that units are important in this equation to ensure the refractive index never falls below the vacuum index of 1.0 at the edge of the component. Here, the node position read from the ANSYS file was used to calculate the nodal refractive-index value, according to Eq. (17).

A uniform random number generator was used to create a variety of ray positions and directions to be launched into the medium. The results of one such ray are shown in Fig. 3. The trajectory of this ray through the FEM volume is shown in Fig. 4. Note that the optical component is discretized into four element layers. At each of the three element interfaces, an element transformation is performed.

By comparing these trajectories to those obtained via direct integration of Eq. (6) using the closed-form gradient-index Eq. (17), the error in the FE solution may be obtained. For the given ray, the error plot is shown in Fig. 5. Since the refractive index of Eq. (17) is not dispersive, it makes little sense to report the error in fractions of a wavelength. However, note that the final error is less than 2 nm in each axis, and this is much smaller than 1/100th of a wave at the shortest wavelengths of the visible spectrum.

By observation, the increasing trend of the error does elicit some concern for generalized application of the method to other component geometries. The exponential growth of the error may generate very large errors for thicker components and longer path

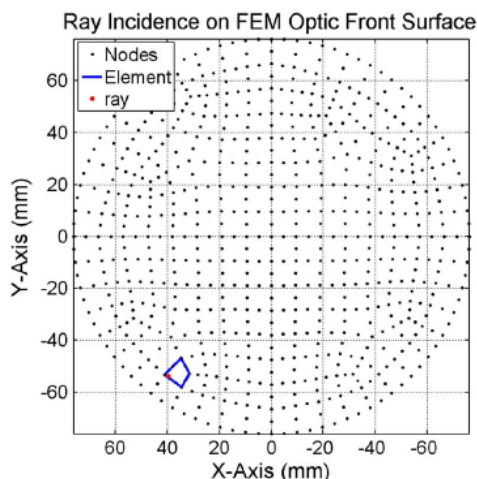


Fig. 3. Ray launch into finite-element volume.

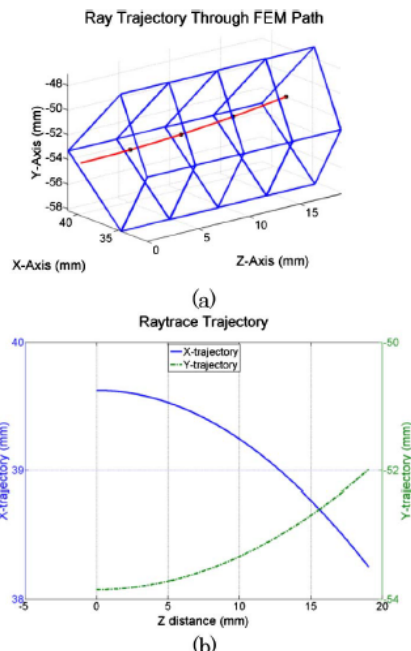


Fig. 4. Ray trajectory (a) through the FEM volume and (b) in X and Y axes along the Z thickness of the optical component.

lengths. Initial findings on limitations of the FE method have shown that the error is cumulative from element to element. Also, the error is dependent upon the severity of the gradient index experienced by the ray along its trajectory. For instance, normal incidence rays launched near the center of the radial gradient component experience almost no gradient and exhibit very little error, while transverse rays incident at the edge of the window experience an extremely severe gradient with very large errors. By jointly considering these two effects, an error-bounding metric may be postulated based upon the severity of the gradient contained in any single element. This, in turn, leads the modeler to discretize an optical component such that the index gradient is

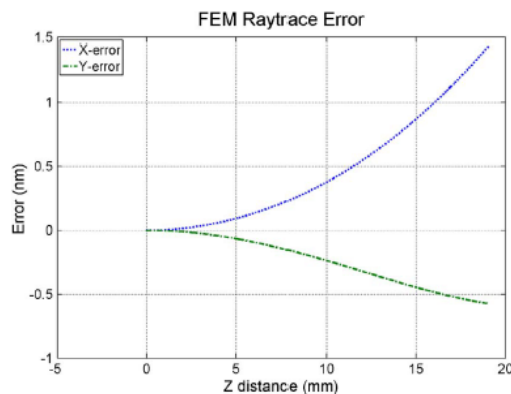


Fig. 5. Error in FE ray trace shown in Fig. 4, compared to closed form solution.



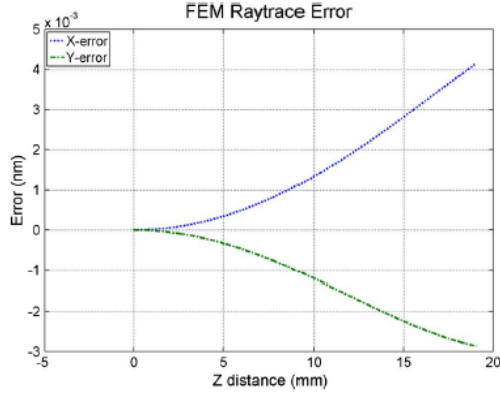


Fig. 6. Error in high-density mesh FE ray trace, compared to closed form solution.

minimized across all the contained elements. If the gradient is deemed to be sufficiently large, the volume can be meshed to a finer resolution, thereby decreasing the gradient across any given element.

To test this theory, the optical window was remeshed to a much finer resolution, creating eight element layers across the thickness of the component [as opposed to the four layers shown in Fig. 4(a)]. Using the same gradient Eq. (17), the FE ray trace was repeated for the same incident ray. The resulting error for the fine-mesh ray-trace solution, in comparison to the closed-form solution, is shown in Fig. 6. Note that while the error is still increasing, the overall magnitude has been reduced by 2.5 orders of magnitude, lending credence to the postulated assertion regarding mesh density. Since increased mesh density carries a cost in data storage and computational burden, effort should be made to define a bounding error metric that can be used to guide decisions regarding appropriate mesh density.

## 5. Conclusions

By employing the full finite-element construct for tracing rays through a gradient-index medium, an instantaneous ray path may be derived, eliminating the need for an intermediate processing step to integrate FEM data with an optical ray-trace package. This ability enables rapid query of optical performance over a number of specific parameters, including field angle, wavelength, and misalignment impacts of any upstream components. While the approach does contain inherent errors due to the limitations of interpolation shape functions and their derivatives, these errors can be managed to an acceptable level by discretizing the component to a sufficient resolution. Moreover, it is conceivable that an error-bounding metric may be defined, based upon the maximum gradient across any given element in the volume, such that the errors are guaranteed not to exceed an upper limit. Such a metric may be used to refine the mesh to a sufficient resolution to minimize errors to an acceptable level.

## Appendix A: HEX20 Finite-Element Construct

A normalized finite element consists of a local coordinate frame, a node-numbering scheme defined within that frame, shape functions used to interpolate nodal parameters, and shape-function derivatives used to compute the Jacobian and gradients of the nodal parameters. In general, all finite-element types carry a specific construct and methodology to accomplish these computations. While this study is limited to the use of solid hexahedral elements with 20 nodes, the approach may easily be generalized to other element types using the methodologies found in the literature. It is noted, however, that for the application considered herein, the use of lower-order elements may represent some compromise in the geometric representation of the optic as well as the inter-element gradients appearing in Eq. (4).

### 1. Master Element Coordinate Frame

Hexahedral solid elements utilize a local master element coordinate frame with three orthogonal axes, typically denoted  $(\xi, \eta, \zeta)$ . Note that the element is a perfect cube in the local coordinate frame, bounded in the range  $[-1, +1]$  in each axis, as shown in Fig. 1(b).

### 2. Node Numbering

In order to maintain consistency of numbering and methodologies between elements of the same family, node ordering is first defined for the lowest-order element of the class containing the least number of nodes, and then extended to the higher-order elements of the class. For the hexahedral element, the lowest-order element contains eight nodes—one located at each corner. This type of element provides only linear interpolation methods between each node. To obtain higher-order interpolation, it is necessary to include additional nodes along each “edge” of the element, commonly referred to as “mid-side” nodes. The HEX20 element, as shown in Fig. 1, contains 20 nodes: eight corner nodes and 12 additional mid-side nodes along each edge. The numbering scheme is listed in Table 1.

Note that the first eight nodes define the corners of the element, just as for the HEX8 element. The first four nodes lie in the  $\zeta = -1$  plane, while the second

Table 1. Node Numbering for a HEX20 Element

Node No.	$(\xi, \eta, \zeta)$	Node No.	$(\xi, \eta, \zeta)$
1	$(-1, -1, -1)$	11	$(0, +1, -1)$
2	$(+1, -1, -1)$	12	$(-1, 0, -1)$
3	$(+1, +1, -1)$	13	$(0, -1, +1)$
4	$(-1, +1, -1)$	14	$(+1, 0, +1)$
5	$(-1, -1, +1)$	15	$(0, +1, +1)$
6	$(+1, -1, +1)$	16	$(-1, 0, +1)$
7	$(+1, +1, +1)$	17	$(-1, -1, 0)$
8	$(-1, +1, +1)$	18	$(+1, -1, 0)$
9	$(0, -1, -1)$	19	$(+1, +1, 0)$
10	$(+1, 0, -1)$	20	$(-1, +1, 0)$





four lie in the  $\zeta = +1$  plane. The mid-side nodes are numbered sequentially in a similar fashion, with the last four nodes lying in the  $\zeta = 0$  plane.

### 3. Shape Functions

Shape functions are used to interpolate nodal parameters at points  $(\xi, \eta, \zeta)$  that lie between nodes within the element. Specifically, they are used to calculate a weight value that is applied to the subject parameter of each node, in accordance with the proximity of the  $(\xi, \eta, \zeta)$  point of interest. The shape functions of a HEX20 element are listed in Table 2.

By multiplying the calculated weight value of each node by the subject parameter of that node and summing over the set of all nodes, the interpolated value is derived. This is shown mathematically in Eq. (A1):

$$p(\xi, \eta, \zeta) = \sum_{i=1}^{20} N_i p_i, \quad (\text{A1})$$

where  $N_i$  is the  $i$ th shape function, calculated at  $(\xi, \eta, \zeta)$ , and  $p_i$  is the value of the subject parameter  $p$  at node  $i$ . This method of interpolation can be applied to any parameter that is defined at each node within the element (e.g., position  $x, y$ , or  $z$ , in physical coordinates, temperature, refractive index, etc.).

Note that for points  $(\xi, \eta, \zeta)$  that coincide exactly with a node position, the shape functions collapse to yield a weight of one at that node, and zero for all other node weights. In this regard, the shape functions maintain a consistent method of interpolation throughout the element volume, including the actual node positions.

Table 2. Shape Functions of a HEX20 Element

Shape Function
$N_1 = -\frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)(2+\xi+\eta+\zeta)$
$N_2 = -\frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)(2-\xi+\eta+\zeta)$
$N_3 = -\frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)(2-\xi-\eta+\zeta)$
$N_4 = -\frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)(2+\xi-\eta+\zeta)$
$N_5 = -\frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)(2+\xi+\eta-\zeta)$
$N_6 = -\frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)(2-\xi+\eta-\zeta)$
$N_7 = -\frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)(2-\xi-\eta-\zeta)$
$N_8 = -\frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)(2+\xi-\eta-\zeta)$
$N_9 = \frac{1}{4}(1-\xi)(1+\xi)(1-\eta)(1-\zeta)$
$N_{10} = \frac{1}{4}(1+\xi)(1-\eta)(1+\eta)(1-\zeta)$
$N_{11} = \frac{1}{4}(1+\xi)(1-\xi)(1+\eta)(1-\zeta)$
$N_{12} = \frac{1}{4}(1-\xi)(1-\eta)(1+\eta)(1-\zeta)$
$N_{13} = \frac{1}{4}(1-\xi)(1+\xi)(1-\eta)(1+\zeta)$
$N_{14} = \frac{1}{4}(1+\xi)(1-\eta)(1+\eta)(1+\zeta)$
$N_{15} = \frac{1}{4}(1-\xi)(1+\xi)(1+\eta)(1+\zeta)$
$N_{16} = \frac{1}{4}(1-\xi)(1-\eta)(1+\eta)(1+\zeta)$
$N_{17} = \frac{1}{4}(1-\xi)(1-\eta)(1-\zeta)(1+\zeta)$
$N_{18} = \frac{1}{4}(1+\xi)(1-\eta)(1-\zeta)(1+\zeta)$
$N_{19} = \frac{1}{4}(1+\xi)(1+\eta)(1-\zeta)(1+\zeta)$
$N_{20} = \frac{1}{4}(1-\xi)(1+\eta)(1-\zeta)(1+\zeta)$

### 4. Jacobian Matrix

The Jacobian matrix is defined as

$$\mathcal{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}. \quad (\text{A2})$$

Derivation of this matrix in the FE construct is accomplished by implementing Eq. (A1) and replacing the parameter  $p$  with the physical coordinates  $(x, y, z)$ :

$$\begin{aligned} x(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) x_i \\ y(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) y_i \\ z(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) z_i \end{aligned}$$

In matrix notation, this can be written as

$$\begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} N_1 & N_2 & \cdots & N_{20} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{20} & y_{20} & z_{20} \end{bmatrix}.$$

And in this form, it is clear that the only components that have dependency on  $(\xi, \eta, \zeta)$  are the shape functions  $N_i$ . Indeed, the node positions are constant. Therefore, the Jacobian matrix derivation becomes

$$\mathcal{J} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_{20}}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_{20}}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \cdots & \frac{\partial N_{20}}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{20} & y_{20} & z_{20} \end{bmatrix}. \quad (\text{A3})$$

Equation (10) also requires the derivative of the Jacobian w.r.t. “pseudo-time.” This is accomplished by extending the approach in Eq. (A3), where

$$\frac{d\mathcal{J}}{dt} = \frac{d}{dt} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_{20}}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_{20}}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \cdots & \frac{\partial N_{20}}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{20} & y_{20} & z_{20} \end{bmatrix}. \quad (\text{A4})$$

Again, the node positions are constant, so the time derivative only applies to the matrix of shape-function derivatives. The chain rule may be applied to this matrix derivative to further expand its components into a form that is dependent on the second derivatives of the shape functions and the “pseudo-velocity” in the local coordinate frame. It is important to note that for elements with only first-order shape functions, the second derivatives are zero and, hence, so is the rate of change of the Jacobian. The second derivatives of the HEX20 shape functions





in  $(\xi, \eta, \zeta)$  yield a large, segmented matrix with 180 components. The final form of the derivative of the Jacobian is shown in Eq. (A5):

$$\frac{d\mathcal{J}}{dt} = \left\{ \begin{array}{l} \left[ \begin{array}{c} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{array} \right]^T \left[ \begin{array}{ccc} \frac{\partial^2 N_1}{\partial \xi^2} & \frac{\partial^2 N_2}{\partial \xi^2} & \dots & \frac{\partial^2 N_{20}}{\partial \xi^2} \\ \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \frac{\partial^2 N_2}{\partial \xi \partial \eta} & \dots & \frac{\partial^2 N_{20}}{\partial \xi \partial \eta} \\ \frac{\partial^2 N_1}{\partial \xi \partial \zeta} & \frac{\partial^2 N_2}{\partial \xi \partial \zeta} & \dots & \frac{\partial^2 N_{20}}{\partial \xi \partial \zeta} \end{array} \right] \\ \left[ \begin{array}{c} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{array} \right]^T \left[ \begin{array}{ccc} \frac{\partial^2 N_1}{\partial \eta^2} & \frac{\partial^2 N_2}{\partial \eta^2} & \dots & \frac{\partial^2 N_{20}}{\partial \eta^2} \\ \frac{\partial^2 N_1}{\partial \eta \partial \xi} & \frac{\partial^2 N_2}{\partial \eta \partial \xi} & \dots & \frac{\partial^2 N_{20}}{\partial \eta \partial \xi} \\ \frac{\partial^2 N_1}{\partial \eta \partial \zeta} & \frac{\partial^2 N_2}{\partial \eta \partial \zeta} & \dots & \frac{\partial^2 N_{20}}{\partial \eta \partial \zeta} \end{array} \right] \\ \left[ \begin{array}{c} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{array} \right]^T \left[ \begin{array}{ccc} \frac{\partial^2 N_1}{\partial \zeta^2} & \frac{\partial^2 N_2}{\partial \zeta^2} & \dots & \frac{\partial^2 N_{20}}{\partial \zeta^2} \\ \frac{\partial^2 N_1}{\partial \zeta \partial \eta} & \frac{\partial^2 N_2}{\partial \zeta \partial \eta} & \dots & \frac{\partial^2 N_{20}}{\partial \zeta \partial \eta} \\ \frac{\partial^2 N_1}{\partial \zeta \partial \xi} & \frac{\partial^2 N_2}{\partial \zeta \partial \xi} & \dots & \frac{\partial^2 N_{20}}{\partial \zeta \partial \xi} \end{array} \right] \end{array} \right\} \times \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{20} & y_{20} & z_{20} \end{bmatrix}. \quad (\text{A5})$$

## 5. Gradients

Parameter gradients in the local coordinate frame within the FE volume are derived in a similar fashion to the Jacobian matrix. Namely,

$$\left[ \begin{array}{c} \frac{\partial p}{\partial \xi} \\ \frac{\partial p}{\partial \eta} \\ \frac{\partial p}{\partial \zeta} \end{array} \right] = \left[ \begin{array}{ccc} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_{20}}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_{20}}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \dots & \frac{\partial N_{20}}{\partial \zeta} \end{array} \right] \left[ \begin{array}{c} p_1 \\ p_2 \\ \vdots \\ p_{20} \end{array} \right]. \quad (\text{A6})$$

This can be applied to any parameter with values defined at each node (temperature, refractive index, etc.).

## 6. Element Generalization

The approaches outlined here are applicable to any finite-element type, within the limits of the specific type chosen. For example, first-order elements have linear shape functions, and the second derivatives collapse to zero. While the construct still applies, the component of Eq. (10) that requires the derivative of the Jacobian will vanish, which will undoubtedly introduce error into the ray-trace calculation. Even so, any second- (or higher-) order element (e.g., tetrahedrons) can be accommodated. The shape functions of common element types are available in the open literature.

## Appendix B: Physical Coordinates to Local Master Element Coordinates Transformation

With respect to interpolating position, the FE construct provides a direct method to convert from the

local coordinate frame to the physical coordinate frame. This conversion is based on the shape functions, as shown in Appendix A, and is typically non-linear in nature. Inverting the transformation to find local coordinates from a given set of physical coordinates can be ill posed and introduce an unacceptable error into the derived local coordinates.

Since the ray-trace algorithm requires a reliable conversion from the given physical ray intersection point to the local coordinate frame of the incident element, a more rigorous algorithm must be applied. Several methods can be found in open-source literature. The ultimate goal of any conversion algorithm is to yield local element coordinates that, when applied to the shape-function interpolation, subsequently yield the original physical coordinates.

The method used for this study is a gradient-descent approach. A cost function is derived from the squared error of the interpolated value from the original coordinates, i.e.,

$$\text{err} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_1 & x_2 & \dots & x_{20} \\ y_1 & y_2 & \dots & y_{20} \\ z_1 & z_2 & \dots & z_{20} \end{bmatrix} \begin{bmatrix} N_1(\xi, \eta, \zeta) \\ N_2(\xi, \eta, \zeta) \\ \vdots \\ N_{20}(\xi, \eta, \zeta) \end{bmatrix}, \quad (\text{B1})$$

$$\text{cost} = |\text{err}|^2 = \text{err}^T * \text{err}. \quad (\text{B2})$$

The gradient of the cost may be calculated directly, to yield

$$\nabla \text{cost} = 2(\nabla \text{err}) * \text{err} = -2\mathcal{J} * \text{err}. \quad (\text{B3})$$

These formulas can be incorporated into any gradient-descent minimization algorithm. The method used for this study was a conjugate-gradient minimization, which can be found in [14].

To reduce the computational burden, note that the point at which the ray intersects the element is a face of the element. As such, the corresponding intersection plane in local coordinates has a value of either +1, or -1. Therefore, this coordinate is already known in the local coordinate frame. The other two coordinates may be found by reducing the problem to that of intersection with a shell (or plate) element that defines the surface of incidence. For a HEX20 element, the surface can be described by an eight-node quadrilateral, or QUAD8 shell element. The same approach outlined in Eqs. (B1)–(B3) may then be employed using only eight nodes and shape functions, with dependency on only the two remaining local coordinate axes.

## Appendix C: Inter-Element Transformation Matrix

A simple inter-element transformation can be derived by exploiting the node-numbering scheme of the neighboring elements, assuming the elements are members of a common family (e.g., HEX or



Table 3. Node Local Coordinates for Element 1

Axis	Node			
	2	3	7	6
$\xi$	+1	+1	+1	+1
$\eta$	-1	+1	+1	-1
$\zeta$	-1	-1	+1	+1

Table 4. Node Local Coordinates for Element 2

Axis	Node			
	2	3	4	1
$\xi$	+1	+1	-1	-1
$\eta$	-1	+1	+1	-1
$\zeta$	-1	-1	-1	-1

TET elements). The technique involves collecting the local coordinates of the nodes contained in the common face in each element, respectively. To illustrate, we use the previous example of a ray exiting from element 1 and entering element 2, as shown in Fig. 2. Using the first four nodes [2, 3, 7, 6] from the common face in element 1, which correlate to nodes [2, 3, 4, 1] of element 2, we collect the node positions in each local coordinate frame, as listed in Table 1 of Appendix A. This is illustrated in Tables 3 and 4.

A basis is then formed from vectors derived from these node local coordinates, in each element. The first vector is the definition of the common plane, found by taking the mean of each row of the tables above:

$$v_{1a} = \text{mean} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}_{\mathcal{E}1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};$$

$$v_{2a} = \text{mean} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}_{\mathcal{E}2} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

The remaining two vectors of the basis are computed from orthogonal edges of the common face:

$$v_{1b} = \frac{\text{Node3}_{E11} - \text{Node2}_{E11}}{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

$$v_{2b} = \frac{\text{Node3}_{E12} - \text{Node2}_{E12}}{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

$$v_{1c} = \frac{\text{Node6}_{E11} - \text{Node2}_{E11}}{2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$v_{2c} = \frac{\text{Node1}_{E12} - \text{Node2}_{E12}}{2} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

These vectors are then assembled into the respective bases to form

$$\mathcal{B}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \mathcal{B}_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

Since the ray is exiting element 1 and entering element 2, the transformation matrix is derived in the following manner:

$$T = \mathcal{B}_1^{-1} \mathcal{B}_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

In the common plane shared by the two elements, the coordinates of the ray intersection are converted from the local coordinates of the first element to the local coordinates of the second element by this transformation:

$$\begin{bmatrix} \xi_2 \\ \eta_2 \\ \zeta_2 \end{bmatrix} = T \begin{bmatrix} \xi_1 \\ \eta_1 \\ \zeta_1 \end{bmatrix}.$$

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. (SAND2014-2030J).

## References

1. L. Montagnino, "Ray tracing in inhomogeneous media," *J. Opt. Soc. Am.* **58**, 1667-1668 (1968).
2. E. W. Marchand, "Ray tracing in gradient-index media," *J. Opt. Soc. Am.* **60**, 1-2 (1970).
3. A. Sharma, D. V. Kumar, and A. K. Ghatak, "Tracing rays through graded-index media: a new method," *Appl. Opt.* **21**, 984-987 (1982).
4. S. Dorić, "Ray tracing through gradient-index media: recent improvements," *Appl. Opt.* **29**, 4026-4029 (1990).
5. V. Genberg, "Ray tracing from finite element results," *Proc. SPIE* **1998**, 72-82 (1993).
6. B. Richerzhagen, "Finite element ray tracing: a new method for ray tracing in a gradient-index media," *Appl. Opt.* **35**, 6186-6189 (1996).
7. M. Epstein, D. Peter, and M. A. Slawinski, "Combining ray-tracing techniques and finite-element modelling in deformable media," *Q. J. Mech. Appl. Math.* **65**, 87-112 (2012).
8. A. Gatej, J. Wasselowski, and P. Loosen, "Thermo-optical (TOP) analysis by coupling FEM and ray tracing," *Proc. SPIE* **8429**, 84290E (2012).
9. Zemax User's Manual, Release 12, Radiant Zemax LLC, 6 April 2012, p. 346.
10. CODE V Macro-PLUS Reference Manual, Version 10.6, Synopsis, September 2013, p. 231.
11. M. Born and E. Wolf, *Principles of Optics*, 6th ed. (Pergamon, 1980), p. 122.
12. K. B. Doyle, V. L. Genberg, and G. L. Michels, *Integrated Optomechanical Analysis*, 2nd ed. (SPIE, 2012), pp. 10-17.
13. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method: Its Basis and Fundamentals*, 6th ed. (Butterworth-Heinemann, 2005), pp. 146-147.
14. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C—The Art of Scientific Computing*, 2nd ed. (Cambridge University, 1992), pp. 420-425.

## APPENDIX B: ANSYS ANALYSIS AND MODEL PREPARATION FOR OUTPUT TO ZEMAX

Eric Couphos (2617)

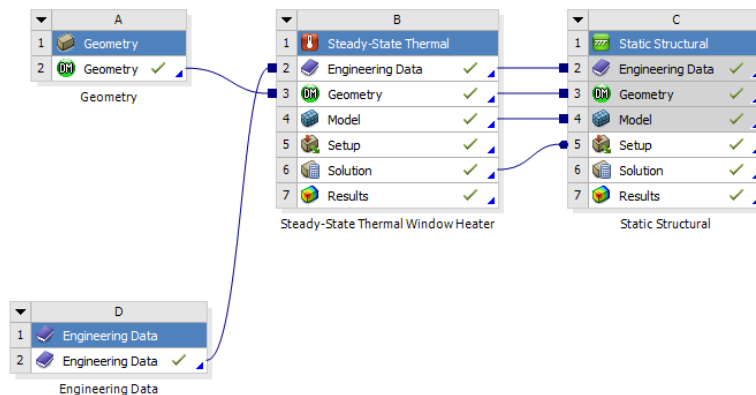
### Introduction

ANSYS is a finite element analysis (FEM) package used to model and analyze thermal and structural loads on systems. This appendix will outline the process used to prepare the ANSYS model and export analysis results into a generic form that can be imported into Zemax for further optical analysis. This allows the user to use Zemax to analyze the optical effects of thermal and structural loads on an optical system.

The output file includes thermal and structural results for all relevant nodes, as well as a list of nodes associated with each optical surface and a list of element connectivity.

### ANSYS Model Setup

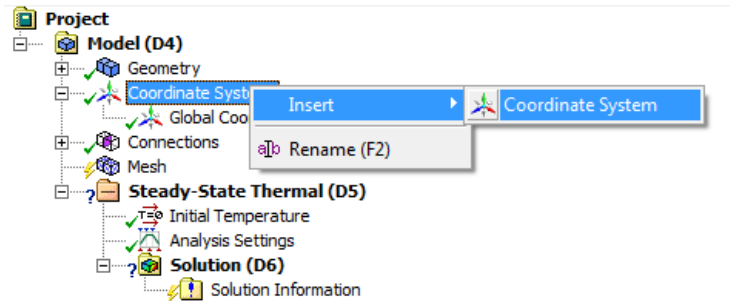
This process is designed to output the results from a sequential thermal and structural analysis within ANSYS. An example project schematic is shown in Figure 1.



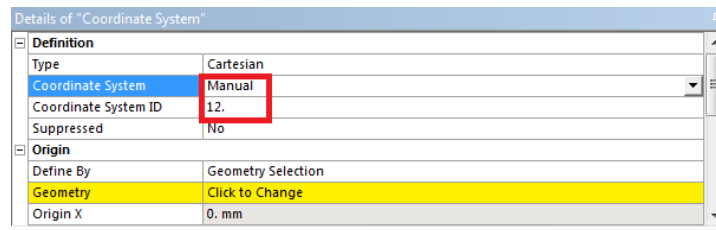
**Figure 1: Ansys Project Schematic**

### *Output Coordinate System*

The output coordinate system must be manually set for the optics in the analysis. The coordinate system should typically be set at the center vertex of surface 1 of the optic, although this may vary depending on the optical assembly. Figure 2 shows how to insert a new coordinate system. The coordinate system type must be changed from 'Program Controlled' to 'Manual' and the coordinate system ID must be set to 12 as shown in Figure 3.



**Figure 2: Insert Coordinate System**



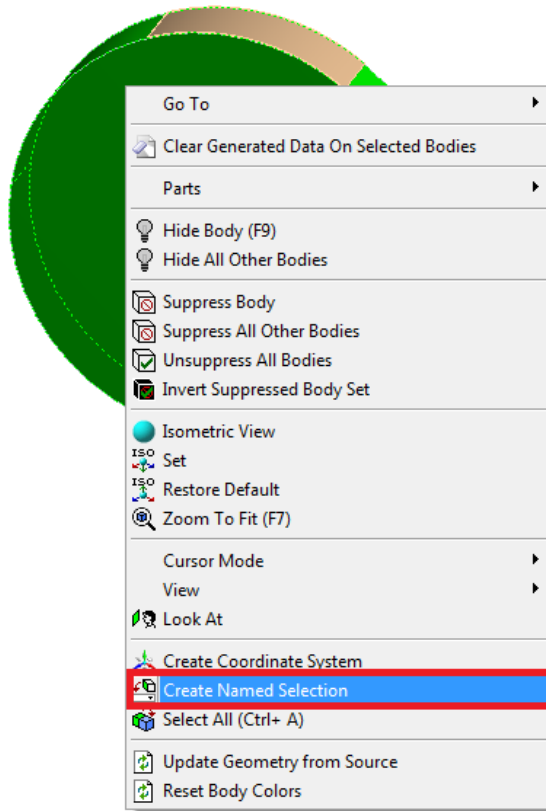
**Figure 3: Set Coordinate System ID**

### *Model Mesh*

Currently the ray trace algorithm can only support 20 node hexahedral (HEX20) elements. The model mesh in Ansys must be configured to mesh the optical elements with only HEX20 elements. This setup will be unique for different optical element geometries and may require slicing or otherwise modifying the geometries in Ansys Design Modeler. Future support for other elements types may be implemented in the future.

### *Named Selections*

Named selections are used to control which nodes are used for data output. The nodal information is only needed for nodes associated with optical elements. The code will output information for all nodes in a named selection called `_optics`. This named selection must be created by the user and titled `_optics`. Figure 4 shows how to create a named selection.



**Figure 4: Create Named Selection**

Named selections must also be created for the front and rear surfaces of optics and labelled *surface\_1* and *surface\_2* respectively. If there are multiple optical elements, elements should be labelled *lens\_1*, *lens\_2*, etc. Additional optical surfaces should be labelled *surface\_3*, *surface\_4*, etc.

### *Output Solution Results*

The code used to output the ANSYS analysis results is inserted as an APDL command within the solution dropdown in the project tree. Two different sets of code are used; one in the steady-state thermal solution and one in the static structural solution. Figure 5 shows how to add a command object, and Figure 6 shows how to import the code from a text file.



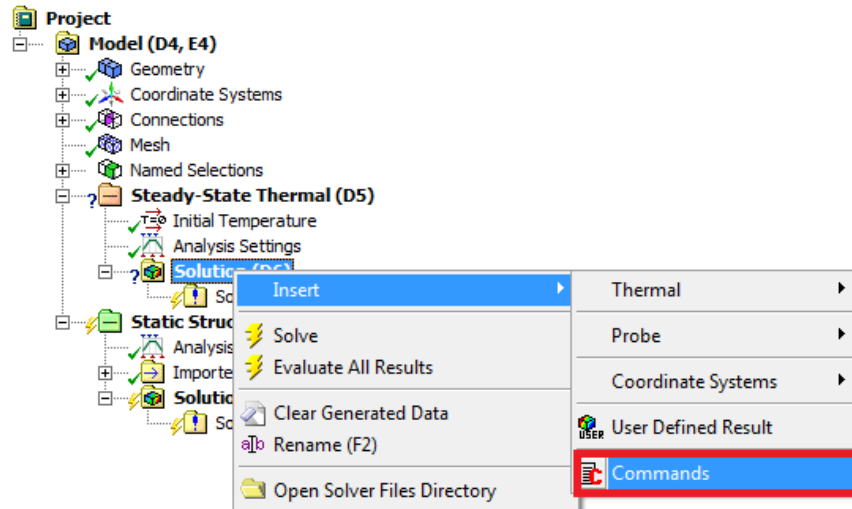


Figure 5: Insert Solution Command

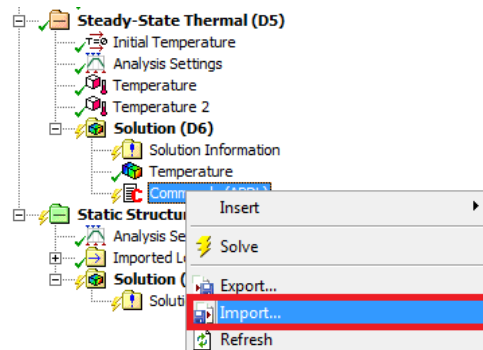


Figure 6: Import Code

The output file location and file name are set in the code used to export thermal results. The output file location must be set to a local drive and not a network drive. An excerpt of this code is shown in Figure 7.

The analysis step number to be exported is also set within this code, as shown in Figure 7.

```

1  !   Prepared by: Eric Couphos, Org 2617. August 29th, 2014
2
3  !   Commands inserted into this file will be executed immediately after the Ansys /POST1 command.
4  !   THIS FILE WILL OUTPUT A HEADER CONSISTING OF USERNAME, FILENAME, UNITS, AND
5  !   NODE TEMPERATURE AND INITIAL POSITION
6
7  !   PLEASE SET OUTPUTLOC PARAMETER TO DESIRED OUTPUT DESTINATION & FILENAME (EXCLUDING .TXT FROM FILENAME)
8  !   EXAMPLE: outputloc(1)='C:\Users\ejcough\Desktop\sample_output'
9  *DIM,outputloc,STRING,248,1
10 outputloc(1)='C:\Users\ejcough\Desktop\sample_output'
11
12 !SET DATA SET (analysis timestep) TO BE READ FROM RESULTS FILE
13 !timestep=last seems to not be working and defaults to timestep=1?
14 timestep=1
15 set,timestep

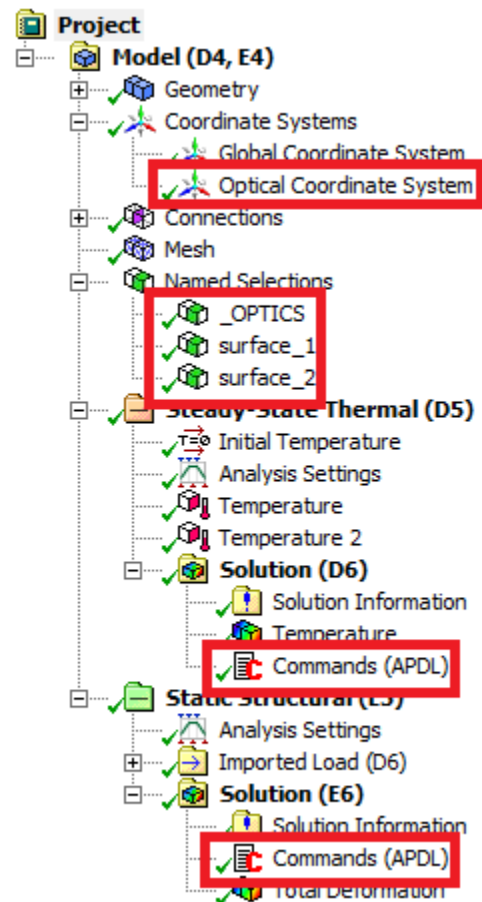
```

Figure 7: Thermal Results Output Code Snippet



## Summary

The process outlined above will allow the user to export FEM analysis results to a text file in a generic format that can be subsequently imported into Zemax for further analysis. The completed project tree must include the elements noted in Figure 8. These elements include a new coordinate system, named selections, and commands.



**Figure 8: Final Project Tree**

## Ansys Code & Sample Output File Locations

The Ansys code required is currently located in the following folder:  
\\citadel\TOPCode\Analyses\ANSYS-Coupfos\Ansys Commands

Output files are located in the following folder:  
\\citadel\TOPCode\Analyses\ANSYS-Coupfos\Analysis Output TXT Files

PeDaL is a program offered by PADT that is inexpensive and excellent for editing ANSYS code. The code used to export thermal results is titled *140829Ansys\_thermal\_node\_export.txt* and the code used to export structural results and element connectivity is titled *140829Ansys\_structural\_node\_export.txt*.



## APPENDIX C: ZEMAX COMPATIBLE SIERRA MECHANICS FEM MODEL

Sam Subia (1541)

The stated objective of this LDRD is to improve the design of optical lenses. These designs are usually developed using optical design software packages. Here the objective is to include the effects of temperature and deformation obtained from FEM numerical simulations during the design process. One aspect of the project is to enable this information to be imported into the Zemax lens design software. Initial efforts to this end have focused on importing temperature and displacement results from ANSYS Mechanical into Zemax. Since our goal is to apply the methodology with arbitrary FEM results we now seek to extend the import process employed with ANSYS results to include a Sierra Mechanics code.

In demonstrating the import process with results from a different application code it is important that we perform this demonstration using the same model. Hence we begin the exercise by attempting to generate the same FEM mesh. Owing to differences in how FEM models are constructed one discovers that obtaining the same FEM mesh is often difficult. The outline below describes the process of generating an equivalent of the ANSYS meshed discretization.

An initial lens model was constructed using Hex20 elements in the ANSYS Mechanical software. ANSYS Mechanical provides an export capability of the mechanical mesh to that of another ANSYS product. Since ANSYS Mechanical is the only ANSYS product that supports the Hex20 this limitation immediately restricts the achievable geometric resolution of the export mesh. Nonetheless, our effort continued in order to define a methodology that could use to service other application codes as well.

The steps used to obtain an ExodusII model geometry is outlined as follows.

Export mesh from ANSYS:

1. Enable the export of a Fluent mesh in ASCII format from the Model – Export dialog
2. Export the mesh – selecting File – Export
3. Save the project – failure to do will delete the exported mesh – *name.msh* .

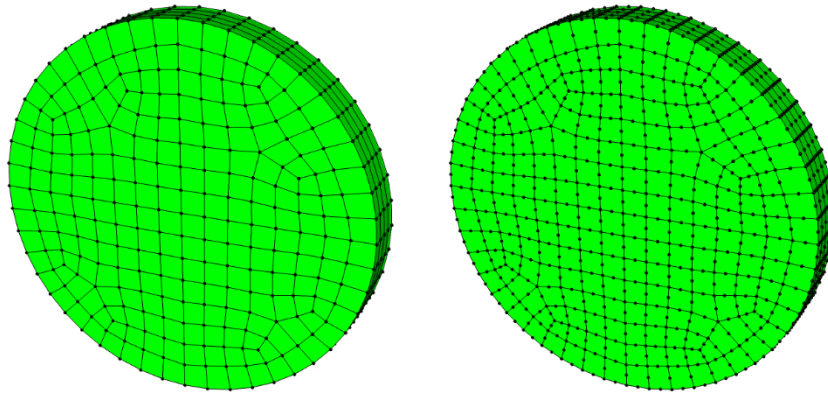
Obtain an ExodusII Mesh:

1. Open the Fluent mesh in Paraview – selecting the “all files” option, then select the Fluent Case Files option
2. Export the ExodusII mesh – select the File – Save Data option write the file as “ExodusII”

Generate a Mesh – the output from Paraview is a HEX8 mesh and we need a HEX20 mesh

1. Import the mesh into Cubit using default options
2. Add side sets as needed
3. Reset the element type – BLOCK # ELEMENT TYPE HEX20  
EXPORT GENESIS “new\_exodus\_file.g”

The original and modified mesh is shown in Figure C.1. At this point we are ready to proceed in developing an equivalent ANSYS model using Sierra Mechanics.



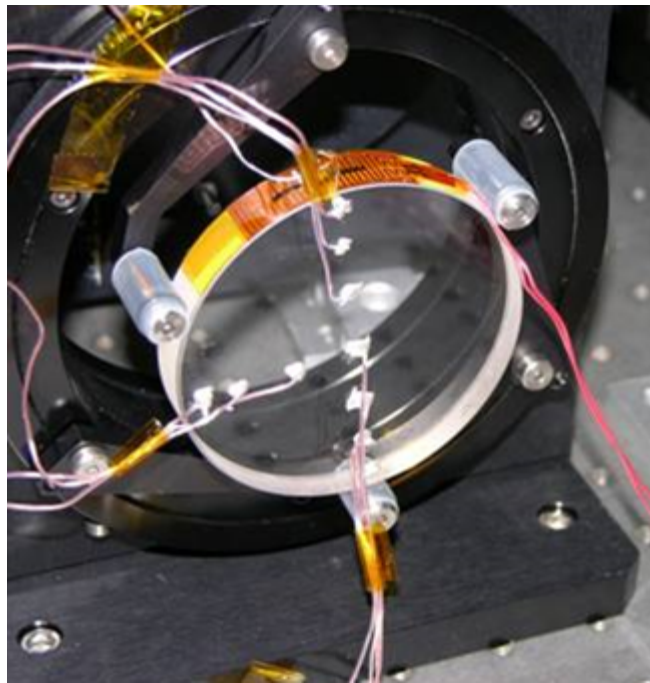
**Figure C.1 Original and modified mesh.**

## APPENDIX D: GRAVITY & MOUNT INDUCED DEFORMATION OF OPTIC DURING THERMAL TESTING

Eric Couphos (2617)

### Introduction

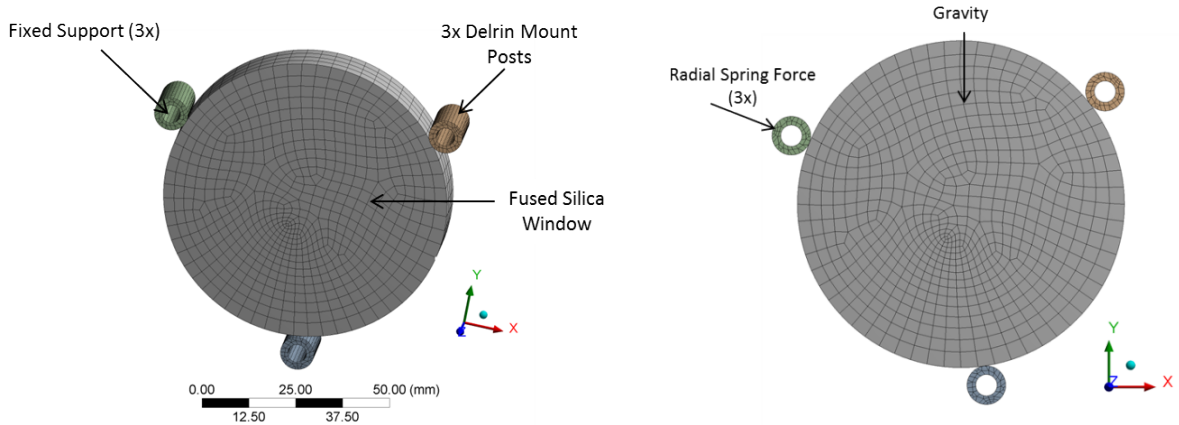
Part of the TOP LDRD consists of thermal testing of a BK7 window and a fused silica window. The 3" diameter, 0.5" thick, windows are heated to create a thermal gradient and measurements of the surface temperature and distortion are obtained with an interferometer, IR camera and thermocouples. The windows are held during testing in a 3 point spring loaded self-centering lens mount shown in Figure 1. FEA analysis was performed to estimate the effects of the spring loaded mount and gravity on the window during testing and determine if these effects should be accounted for during subsequent analysis. Both effects were determined to have a negligible effect on surface deformations and can be ignored in more detailed analysis of the thermal gradient induced deformations of the windows.



**Figure 1: 3 Point Self-Centering Mount**

## Model Setup

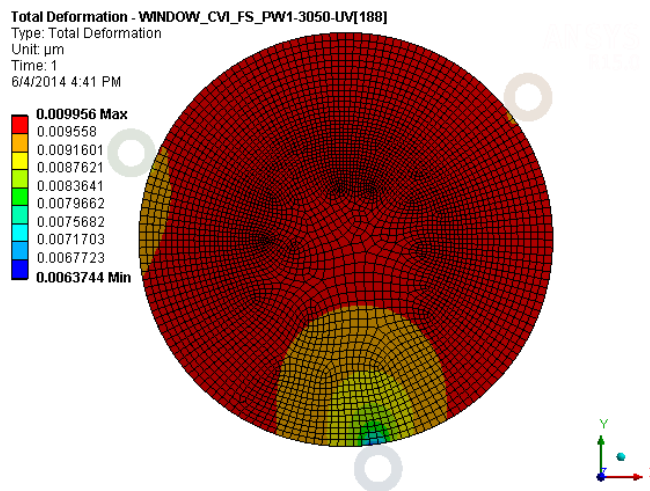
The ANSYS model shown in Figure 2 consists of 3 delrin supports and a fused silica window. BK7 was not modeled as it is stiffer than fused silica and should have smaller deformations. This is a simplified model of the mount used for actual testing. The inner diameters of the delrin supports are fixed and the window is simply supported by these 3 delrin supports. An inward radial force was also included at each of the 3 delrin supports to represent the spring-loaded centering force of the mount. Gravity is acting in the negative Y direction.



**Figure 2: ANSYS Model Setup**

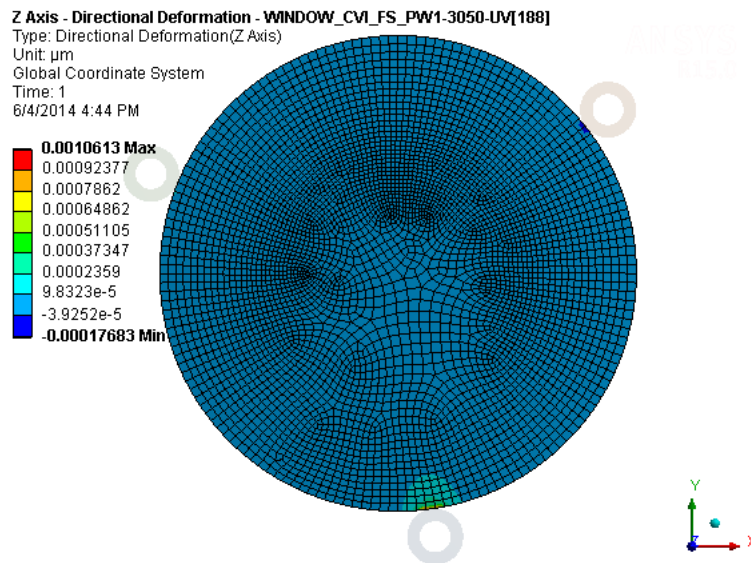
## Analysis Results

Figure 3 shows the total deformation of the window due to gravity and mount stresses. This motion consists primarily of a 9 nanometer (nm) rigid body decenter of the window. This decenter value can be ignored during subsequent analysis as it is a constant value that should not change during testing and is very small compared with the expected deformations of the window during testing.



**Figure 3: Total Window Deformation**

Figure 4 shows the deformation of the front surface in the axial direction. It shows a peak surface deformation of 1 nm which is artificially high due to the simplified contact with the delrin support.



**Figure 4: Front Surface Axial Displacement**

## **Summary**

The effects of the mount and gravity can be ignored as negligible during further detailed analysis of this test setup. A rigid body decenter motion of 9 nanometers and a front surface deformation of 1 nanometer are anticipated based on the results of this analysis. These deformations are very small compared to the expected surface deformation during thermal testing and were not apparent during initial interferometry measurements of the mounted window.





APPENDIX E: SAMPLE FEM FILE FORMAT

USER: ejcough  
FILENAME:140606\_hot\_cold\_optic\_and\_window\_heater--Steady-St  
ANALYSIS DATE (YEAR,MONTH,DAY):20140606 TIME (HOUR,MIN,SEC): 91543

\*\*\*\*\* UNITS \*\*\*\*\*

MPA UNITS SPECIFIED FOR INTERNAL  
LENGTH = MILLIMETERS (mm)  
MASS = TONNE (Mg)  
TIME = SECONDS (sec)  
TEMPERATURE = CELSIUS (C)  
TOFFSET = 273.0  
FORCE = NEWTON (N)  
HEAT = MILLIJOULES (mJ)  
##NODE INITIAL POSITION DATA##  
node\_num x-int y-int z-int  
1149 -12.57674302999999938 -26.615913580000000085 3.17499992325902092  
1150 -12.57674302999999938 -26.615913580000000085 6.34999992325902429  
1151 -12.57674302999999938 -26.615913580000000085 9.52499992325902056  
1152 -11.915785500000000014 -24.87900000999999861 3.17499992325902092  
1153 -11.915785500000000014 -24.87900000999999861 6.34999992325902518  
1154 -11.915785500000000014 -24.87900000999999861 9.52499992325902056  
1155 -25.38368980999999991 11.398416250000000036 3.17499992325902092  
1156 -25.38368980999999991 11.398416250000000036 6.34999992325903584  
1157 -25.38368980999999991 11.398416250000000036 9.52499992325902056  
□ □ □ □

##NODE TEMPERATURE DATA##  
node\_num temp  
1149 27.57372860517499191  
1150 27.61192820336622944  
1151 27.57372860517498836  
1152 27.77501079047443255  
1153 27.81470356205176131  
1154 27.77501079047443255  
1155 37.38867795658788395  
1156 37.50157084519752715  
1157 37.38867795658789106  
□ □

\*\*\*\*\* UNITS \*\*\*\*\*

MPA UNITS SPECIFIED FOR INTERNAL  
LENGTH = MILLIMETERS (mm)  
MASS = TONNE (Mg)  
TIME = SECONDS (sec)  
TEMPERATURE = CELSIUS (C)  
TOFFSET = 273.0  
FORCE = NEWTON (N)  
HEAT = MILLIJOULES (mJ)  
##NODE DISPLACEMENT DATA##  
node\_num x-disp y-disp z-disp  
1149 -0.00032528553669064 0.00103324340611798 -0.00012966135511114  
1150 -0.00032591249659619 0.00103622152979977 0.00000005778089883  
1151 -0.00032528553664935 0.00103324340610450 0.00012977691690881  
1152 -0.00033499980654290 0.00113456576591513 -0.00013300463095579  
1153 -0.00033550922775187 0.00113805574523065 0.00000005778089821  
1154 -0.00033499980650148 0.00113456576590165 0.00013312019275222  
1155 -0.00365868893890354 0.00273095530797711 -0.00035156792954221  
1156 -0.00364982301188921 0.00275027109421990 0.00000005778107272  
1157 -0.00365868893885994 0.00273095530796137 0.00035168349168761  
□ □ □ □

##NODE STRESS/STRAIN DATA##  
node\_num stress strain  
1149 1.27625804495906037 0.00001556414008519  
1150 1.29031133930186570 0.00001573552117407  
1151 1.27625804495906037 0.00001556414008519  
1152 1.25378378415310410 0.00001529013570689  
1153 1.26866699011751249 0.00001547163871389  
1154 1.25378378415310365 0.00001529013570689  
1155 1.21736672234805643 0.00001484606821123  
1156 1.14203439343607038 0.00001392737954120  
1157 1.21736672234805643 0.00001484606821123  
□ □ □ □

```
##NAMED SELECTION NODE DATA##  
SURFACE_1  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
□
```

```
SURFACE_2  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
□
```

LIST ELEMENT TYPES FROM 1 TO 6 BY 1  
1

\*\*\*\*\* ANSYS - ENGINEERING ANALYSIS SYSTEM RELEASE 15.0 \*\*\*\*\*  
ANSYS Mechanical  
00601480 VERSION=WINDOWS x64 09:16:54 JUN 06, 2014 CP= 28.408

140606\_hot\_cold\_optic\_and\_window\_heater--Static Structural (E5)

ELEMENT TYPE 1 IS SOLID186 3-D 20-NODE STRUCTURAL SOLID  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 0 0 0 0 0  
KEYOPT(13-18)= 0 0 0 0 0 0

ELEMENT TYPE 2 IS SOLID186 3-D 20-NODE STRUCTURAL SOLID  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 0 0 0 0 0  
KEYOPT(13-18)= 0 0 0 0 0 0

ELEMENT TYPE 3 IS CONTA174 3D 8-NODE SURF-SURF CONTACT  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 2 1 2 0 5  
KEYOPT(13-18)= 0 0 0 0 0 0

ELEMENT TYPE 4 IS TARGE170 3-D TARGET SEGMENT  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 0 0 0 0 0  
KEYOPT(13-18)= 0 0 0 0 0 0

ELEMENT TYPE 5 IS SURF154 3-D STRUCTURAL SURFACE  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 0 0 0 0 0  
KEYOPT(13-18)= 0 0 0 0 0 0

ELEMENT TYPE 6 IS COMBIN14 SPRING-DAMPER  
KEYOPT( 1- 6)= 0 0 0 0 0 0  
KEYOPT( 7-12)= 0 0 0 0 0 0  
KEYOPT(13-18)= 0 0 0 0 0 0

CURRENT NODAL DOF SET IS UX UY UZ  
THREE-DIMENSIONAL MODEL

##ELEMENT CONNECTIVITY DATA##

elem #	elem_typ	#_node	node_numbers																			
141	1	20	1149	1152	2289	2337	4152	4151	3772	3756	5285	5301	9322	5287	14439	13761	13725	13726	5289	5303	9323	9472
142	1	20	1149	2337	2289	1152	1150	2338	2290	1153	5287	9322	5301	5285	5293	9325	5305	5291	5284	9469	9321	5300
143	1	20	1150	2338	2290	1153	1151	2339	2291	1154	5293	9325	5305	5291	5297	9326	5307	5295	5290	9473	9324	5304
144	1	20	1151	2339	2291	1154	4153	4549	4533	4154	5297	9326	5307	5295	14442	15279	14444	14440	5299	9478	9327	5309
145	1	20	1149	2337	2700	2703	4152	3756	3635	3634	5287	9471	10713	5288	13726	13496	13493	13494	5289	9472	10715	10725
146	1	20	1149	2703	2700	2337	1150	2704	2701	2338	5288	10713	9471	5287	5294	10717	9475	5293	5284	10722	10712	9469
147	1	20	1150	2704	2701	2338	1151	2705	2702	2339	5294	10717	9475	5293	5298	10719	9477	5297	5290	10726	10716	9473
148	1	20	1151	2705	2702	2339	4153	4671	4670	4549	5298	10719	9477	5297	14443	15560	15308	14442	5299	10731	10721	9478
149	1	20	1149	2703	2706	1167	4152	3634	3633	4146	5288	10723	5365	5286	13494	13491	13492	14436	5289	10725	10735	5366
□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

## DISTRIBUTION

1	MS0406	Karl N. Schrader	Org. 5783
1	MS0406	Ronald M. Baker	Org. 5783
1	MS0836	Ronald L. Akau	Org. 1514
1	MS0899	Technical Library	9536 (electronic copy)
1	MS0359	D. Chavez, LDRD Office	1911



